



UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH



Aplicació multiplataforma amb Flutter: PetsWorld

Treball final de grau

Memòria del projecte

Grau en enginyeria informàtica

Enginyeria del Software

Data de defensa: 26 d'octubre de 2020

Directora: Dra. Cristina Gómez Seoane

Autor: Raul Mateo Beneyto

Agraïments

A la meva tutora, Cristina Gómez Seoane, per haver-me guiat en l'elaboració d'aquest treball.

A Dani Rodríguez, per introduir-me l'Enginyeria Informàtica i motivar-me.

A la meva família, per creure sempre en mi.

I a tota la gent que m'ha recolzat i m'ha fet sentir a casa en aquest curiós camí.

Resum

Cada any se sacrifiquen milers d'animals de companyia, només a Espanya. Així mateix, segons el diari 20minutos [3], la directora de la fundació Fundacion Affinity: Isabel Buil, va recordar que l'adopció és "la millor manera de combatre l'abandonament a curt i mitjà termini".

Un altre punt a considerar és l'augment de tràfic il·legal d'animals per internet, una de les principals causes d'abandonament. La raó per la qual es produeix aquest fet és que actualment el mercat de mascotes en Espanya no està suficientment controlat [4].

Així doncs, **aquest projecte té com a objectiu oferir** una plataforma que doni **més visibilitat a les adopcions** d'animal – sigui gossos, gats, rèptils o altres-. El resultat serà una aplicació on podràs contactar amb les protectores més properes, comprar productes relacionats amb els animals de companyia, contractar i oferir serveis, posar en adopció un animal, i més.

Resumen

Cada año se sacrifican más de mil animales de compañía, solo en España. Así mismo, según el diario 20minutos [3], la directora de la Fundacion Affinity: Isabel Buil, recordó que la adopción es "la mejor manera de combatir el abandono a corto y medio plazo".

Otro punto a considerar es el aumento de trafico ilegal de animales por internet, una de las principales causas de abandono. La razón de que se produzcan estos hechos es que actualmente el mercado de mascotas en España no está suficientemente controlado.

Así pues, **este proyecto tiene como objetivo ofrecer** una plataforma que de **más visibilidad a las adopciones de animales** – sean perros, gatos, reptiles o otros-. El resultado será una aplicación donde podrás contactar con las protectoras más cercanas, comprar productos relacionados con los animales de compañía, contractar y ofrecer servicios, poner en adopción un animal, y más.

Abstract

Every year more than a thousand pets are sacrificed, only in Spain. Therefore, according to the diary 20minutos [3], the Fundacion Affinity's director: Isabel Buil, remembered that adoption is "the best way to deal with abandonment short and long term".

Another topic to be aware of is the increase of illegal animal trafficking, one of the main causes of abandonment. The reason for this is that currently there is not much control in the animal market in Spain.

Thus, this project aims to offer a platform to give more visibility to pets' adoption. The final result will be an app where you will be able to contact to your closest shelters, buy animal related products, contract or offer services, put an animal for adoption, and more.

Contingut de la memòria

1	Introducció.....	9
1.1	Marc de treball.....	9
1.2	Contingut de la memòria.....	9
2	Context.....	9
2.1	Formulació del problema.....	9
2.2	Estat de l'art.....	11
2.2.1	Mercat d'aplicacions d'Android: Play Store.....	11
2.2.2	Mercat d'aplicacions de iOS: AppStore.....	11
2.2.3	Taula comparativa entre aplicacions.....	12
2.2.4	Conclusió.....	12
2.3	Actors implicats.....	13
2.3.1	Desenvolupador.....	13
2.3.2	Dissenyador.....	14
2.3.3	Directora.....	14
2.3.4	Protectores d'animals.....	14
2.3.5	Professionals relacionats amb els serveis animal.....	14
2.3.6	Professionals relacionats amb els productes animals.....	14
2.3.7	Particulars.....	14
2.3.8	Mascotes.....	14
3	Abast.....	15
3.1	Objectius genèrics.....	15
3.2	Objectius específics.....	15
3.2.1	Objectius tècnics.....	15
3.2.2	Objectius funcionals.....	16
3.3	Obstacles i riscos.....	16
3.3.1	Limitació temporal en l'entrega.....	16
3.3.2	Inexperiència de les tecnologies.....	17
3.3.3	Inexistència de llibreries.....	17

3.3.4	Coronavirus	17
4	Metodologia i gestió del projecte	17
4.1	Mètode de treball.....	17
4.1.1	Eines de seguiment	19
4.2	Planificació inicial	19
4.2.1	Planificació temporal.....	19
4.2.2	Descripció de les tasques	20
4.2.3	Diagrama de Gantt	22
4.2.4	Gestió del risc: obstacles i plans de mitigació	23
4.3	Gestió econòmica	24
4.3.1	Identificació i estimació dels costos	24
4.3.2	Control de gestió	26
5	Anàlisi de requisits i especificació	26
5.1	Requisits funcionals.....	26
5.1.1	Diagrama genèric de casos d'ús	27
5.1.2	Descripció dels casos d'ús	27
5.2	Requisits no funcionals.....	35
5.3	Model conceptual.....	38
5.3.1	Esquema conceptual de les dades	38
5.3.2	Restriccions d'integritat	39
5.3.3	Descripció de les classes.....	39
6	Disseny	41
6.1	Arquitectura	42
6.1.1	Front-end.....	42
6.1.2	Back-end.....	45
6.1.3	Sistema complet.....	46
6.2	Disseny de l'aplicació (front-end).....	48
6.2.1	Disseny de la UI	48
6.2.2	Mapa navegacional de la UI	55

6.2.3	Disseny intern.....	55
6.3	Disseny del back-end.....	58
6.3.1	Mètodes públics de l'API.....	58
6.3.2	Diagrama de classes del back-end.....	60
6.3.3	Diagrama de seqüència.....	62
6.4	Esquema de la base de dades.....	62
7	Implementació.....	66
7.1	Implementació del front-end.....	66
7.1.1	Versions d'eines.....	66
7.1.2	Filosofia de Widgets.....	67
7.1.3	Tecnologies usades i aplicades.....	67
7.2	Implementació del back-end.....	72
7.2.1	Apollo Platform i Express.....	72
7.2.2	GraphQL Playground.....	72
7.2.3	Servei d'allotjament d'imatges.....	73
7.2.4	Algorisme de paginació.....	73
7.2.5	Base de dades: MongoDB.....	76
7.2.6	Desplegament.....	77
8	Validació i proves.....	77
8.1	Validació manual dels requisits funcionals.....	78
8.2	Estratègia de proves dels requisits no funcionals.....	82
8.3	Validació dels requisits no funcionals.....	83
9	Resultats de la gestió del projecte.....	85
9.1	Resultats a nivell de la metodologia.....	86
9.1.1	Mètodes de treball.....	86
9.1.2	Eines de seguiment.....	86
9.1.3	Mètode de validació.....	87
9.2	Resultats a nivell de la planificació.....	87
9.3	Resultats a nivell de gestió econòmica.....	89

9.4	Sostenibilitat.....	89
9.4.1	Dimensió econòmica	90
9.4.2	Dimensió ambiental	90
9.4.3	Dimensió social.....	91
10	Conclusions	92
10.1	Limitacions i dificultats	92
10.2	Integració de coneixements.....	93
10.3	Justificació de les competències	94
10.4	Treball futur	95
11	Glossari.....	97
11.1	Què és una aplicació o <i>software</i> ?	97
11.2	Què és front-end i back-end?	97
11.3	Què és una API?	97
11.4	Què és UI i UX? [36]	97
11.5	Què és un Framework?	97
11.6	Què és Flutter?	98
11.7	Què és PetsWorld?	98
12	Referències	98
Annex 1.	Diagrama de Gantt	102
Annex 2.	Índex de taules	102
Annex 3.	Índex de il·lustracions.....	104

1 Introducció

1.1 Marc de treball

Aquest projecte és un Treball Final de Grau del grau d'Enginyeria Informàtica impartit per la **Facultat d'Informàtica de Barcelona** de la Universitat Politècnica de Catalunya [1], i més concretament a l'especialització en Enginyeria del Software. Es tracta d'un treballat proposat i realitzat per l'estudiant **Raul Mateo Beneyto**, amb la direcció de la **Prof. Cristina Gómez Seoane**. Tot i que l'objectiu tècnic d'aquest treball és desenvolupar una aplicació mòbil, aquest treball forma part d'un projecte més gran – no documentat- on l'aplicació final englobarà també un portal web.

1.2 Contingut de la memòria

L'objectiu d'aquesta memòria és documentar tots els passos necessaris per a desenvolupar un projecte d'enginyeria del software. Aquest document serveix per a demostrar que l'autor sap aplicar els coneixements adquirits durant tots els cursos universitaris, i sobretot, durant l'especialització de la carrera.

Aquest projecte d'enginyeria requereix de diferents rols, que seran definits i explicats més endavant en l'apartat 2.3.

D'entrada s'estudiarà el context i l'estat actual de l'art. Un cop investigat això, es definirà l'abast del projecte, junt amb la metodologia i el rigor que es farà servir. Seguidament es realitzarà una anàlisi de requisits per a definir els requisits funcionals i els no funcionals de l'aplicació. Posteriorment es dissenyarà l'arquitectura del sistema, que s'anirà completant de manera progressiva a mesura que aquest vagi avançant, junt amb l'apartat d'implementació. En darrer lloc, es validarà que l'aplicació compleixi els requisits definits inicialment i s'estudiarà la sostenibilitat i la gestió del projecte.

2 Context

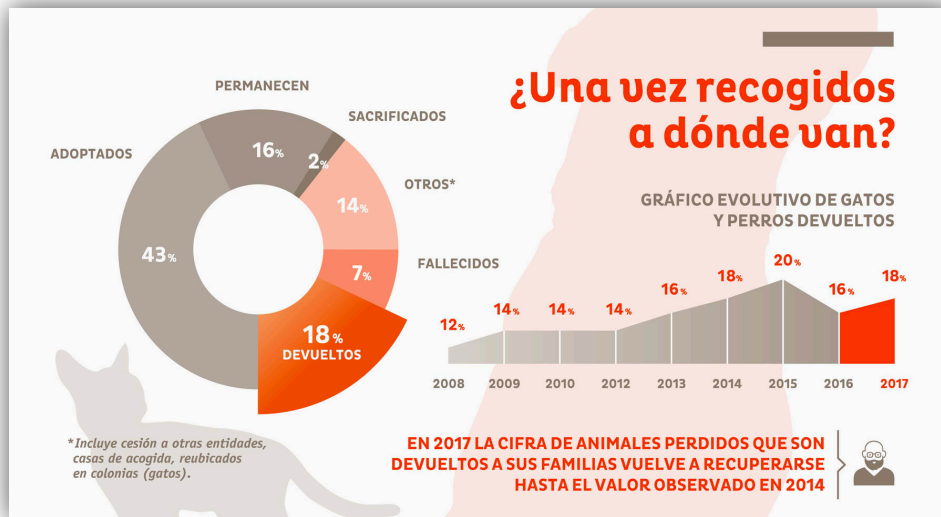
2.1 Formulació del problema

Actualment existeixen diverses preocupacions relacionades amb l'adopció de mascotes. Seguidament es mostraran els problemes més significatius els quals han motivat a la realització d'aquest projecte.

D'entrada, l'abandonament i la pèrdua de gossos i gats constitueixen el principal problema de benestar dels animals de companyia en Espanya. Un estudi de l'any 2018 publicat per Fundacion Affinity¹ [2] mostra que en aquell any es van recollir quasi 140.000 mascotes abandonades, les quals

¹ Fundació creada per Affinity Petcare, multinacional d'aliments de gossos i gats.

menys de la meitat van ser adoptades. Així mateix, com es pot veure a la gràfica de la Il·lustració 1, al voltant de 2.800 d'aquests animals van ser sacrificats. A més, segons el diari 20minutos [3], la directora de la fundació: Isabel Buil, va recordar que l'adopció és "la millor manera de combatre l'abandonament a curt i mitjà termini", i així evitar que aquests nobles animals visquin la resta de la seva vida en una protectora, o pitjor: sent sacrificats.



IL·LUSTRACIÓ 1: GRÀFICA, EN CASTELLÀ, SOBRE EL DESTÍ DELS ANIMALS ADOPTATS EN EL 2017 [2]

Un altre punt a considerar és l'augment de tràfic il·legal d'animals per internet, una de les principals causes d'abandonament. La raó per la qual es produeix aquest fet és que actualment el mercat de mascotes en Espanya no està suficientment controlat [4]. Com a exemple, en plataformes de portals d'anuncis com MilAnuncios² i Vibbo³ es pot comprar fàcilment un cadell de pastor belga on l'única informació que té l'usuari sobre el criador és el seu número de telèfon. A causa d'aquesta facilitat per vendre animals sense cap necessitat d'aportar documents verídics (per validar que, realment, aquest venedor és un criador autoritzat), els veterinaris es troben usualment casos on els animals es posen malalts al cap de poc temps d'haver estat adoptats; conseqüència de no haver estat tractats en bones condicions o de no portar les vacunes recomanades.

En últim lloc, l'adopció per internet actualment està limitada, ja que existeixen poques plataformes destinades a aquest tràmit. En l'apartat 2.2 s'analitzarà el mercat més detalladament.

² Portal d'anuncis de segona mà, <https://www.milanuncios.com/>

³ Un altre portal d'anuncis de segona mà, anteriorment anomenat [segundamano.es](https://www.vibbo.com/); <https://www.vibbo.com/>

En conjunt, el **principal problema que s'intentarà resoldre**, o almenys minimitzar, amb el desenvolupament d'aquest projecte és la **falta de llars per a mascotes**. Sobretot aquelles que actualment viuen en protectores, en cases (on són maltractades i/o descuidades), o en el carrer.

2.2 Estat de l'art

Per justificar el desenvolupament tècnic d'aquest projecte correctament s'ha de realitzar una investigació de solucions similars ja existents. A més, com que aquest treball de final de grau es basa en el desenvolupament d'una aplicació per a mòbil, limitarem la investigació en trobar solucions existents dintre del mercat d'aplicacions d'Android i de iOS.

2.2.1 Mercat d'aplicacions d'Android: Play Store

2.2.1.1 Acoge un perro

Acoge un perro [5] és una aplicació per a Android la qual es basa en dues funcionalitats: adoptar qualsevol animal, principalment gossos, i buscar les gosses més properes de qualsevol ciutat d'Espanya. A part d'aquestes dues principals funcionalitats l'aplicació no ofereix gran cosa més.

2.2.1.2 Miwuki

Miwuki [6] és una aplicació més, també existent en el mercat d'Android. Aquesta també es basa en adoptar qualsevol animal, no només gossos. A diferència de l'anterior aplicació, aquesta permet trametre l'adopció a través de la mateixa aplicació.

2.2.1.3 Chuby

Chuby [7] és una aplicació quasi idèntica a *Acoge un perro*, ja que conté les mateixes funcionalitats.

2.2.1.4 Appets

Appets [8] és una altra aplicació d'Android per a adoptar mascotes. A diferència de les altres aplicacions aquesta si que ofereix anuncis de serveis de veterinària i de passejadors de mascotes.

2.2.2 Mercat d'aplicacions de iOS: AppStore

2.2.2.1 AdoptMe

AdoptMe és la primera aplicació trobada de l'AppStore. Aquesta ofereix la possibilitat d'adoptar una mascota, a la vegada que permet enviar i rebre missatges a la persona o entitat que dona en adopció la mascota.

2.2.2.2 Mascomad

Mascomad és una aplicació desenvolupada per la Comunidad de Madrid. L'objectiu d'aquesta app és poder trobar la teva mascota perduda, notificar sobre animals perduts, i a més et facilita dades sobre gossos en adopcions, i del centre al qual pertanyen.

2.2.3 Taula comparativa entre aplicacions

Després d'investigar les diferents solucions existents actuals, s'ha realitzat una taula comparativa, la Taula 1, per resumir visualment la situació i el context en la que PetsWorld se situa davant de la seva competència directa. A més, s'ha ressaltat en verd les graelles que concorden amb la nostra aplicació.

	Acoge un perro	Miwuki	Chuby	Appets	AdoptMe	Mascomad	PetsWorld
Usuaris	> 500	> 100k	> 10k	> 5k	No info.	No info.	-
Inici de sessió obligatori	No	Si	Si	No	No	No	No
Adopcions	Si	Si	Si	Si	Si	Si	Si
Productes sobre mascotes	No	No	No	No	No	No	Si
Serveis sobre mascotes	No	No	No	Si	No	No	Si
Protectores filtrades per proximitat	No	No	No	No	No	No	Si
Categories per mascotes de diferents espècies	No	No	No	No	No	No	Si
Multiplataforma	No	No	No	No	No	No	Si

TAULA 1: COMPARATIVA ENTRE APLICACIONS EXISTENTS I PETS WORLD

2.2.4 Conclusió

Després d'investigar les aplicacions més significatives dels mercats d'aplicacions, s'ha arribat a la conclusió que, en general, les solucions existents manquen de seccions relacionades en l'àmbit animal; i per tant, no centralitzen el públic objectiu en la seva plataforma. A conseqüència d'això les aplicacions existents perden l'oportunitat d'augmentar les fonts d'ingressos d'aquesta, el que provoca un pobre manteniment del sistema.

Així mateix, gran part d'**aquestes aplicacions mantenen un disseny desactualitzat i poc intuïtiu**, provocant així una experiència d'usuari bastant millorable. Com a exemple, algunes d'aquestes aplicacions requereixen a l'usuari registrar-se per tal de visualitzar el contingut d'aquestes. Aquest requisit és bastant crític, ja que demanar a un nou usuari que es registri en el teu sistema, sense haver vist abans cap altra pantalla o contingut de l'aplicació, pot provocar fàcilment que l'usuari busqui una alternativa.

A més, cap de les alternatives analitzades anteriorment tenen categories de mascotes per a diferents espècies. A causa d'això els usuaris que posseeixen animals domèstics que no siguin gossos queden exclosos de les aplicacions relacionades amb l'àmbit animal. Per aquesta raó, **en PetsWorld s'acceptarà anuncis de mascotes de diferents espècies**, com gats, ocells, rèptils, etc.

En conseqüència a aquests inconvenients, i del fet que cap aplicació d'aquestes sigui de codi obert, l'autor ha decidit que la **millor solució és desenvolupar una aplicació nova des de zero**. A continuació s'anomenaran els avantatges que aporta aquesta solució.

Primerament, es podrà enfocar i desenvolupar l'aplicació perquè sigui **multiplataforma: per a iOS i Android**. En segon lloc, existeix l'oportunitat de poder crear un **disseny nou i innovador**, i així millorar notablement l'experiència d'usuari de les solucions existents. Per acabar, al desenvolupar tota l'aplicació el mateix equip, comporta una millora de la qualitat del codi pel posterior manteniment d'aquest; i a la vegada no dependre de sistemes ja implementats.

En resum, la millor opció disponible, tenint en compte els sistemes existents per a adoptar mascotes a través del mòbil, és desenvolupar una aplicació des de zero per a Android i iOS, valorant significativament l'UI⁴ i l'UX⁵ d'aquesta i aportant noves seccions i funcionalitats.

2.3 Actors implicats

A continuació es mostrarà quins són els principals actors implicats en el desenvolupament i funcionament del sistema.

2.3.1 Desenvolupador

És la persona que s'encarrega de definir l'arquitectura de l'aplicació, implementar, i desplegar l'aplicació. A més també és responsable de provar i mantenir la validesa del codi. En aquest cas el rol de desenvolupador l'ocuparà l'autor de la documentació.

⁴ Interfície d'usuari

⁵ Experiència d'usuari

2.3.2 Dissenyador

Rol encarregat de dissenyar la interfície d'usuari per tal d'assegurar una correcta i gratificant experiència d'usuari. Aquest rol també serà ocupat per l'autor d'aquest treball de final de grau.

2.3.3 Directora

Dra. Cristina Gómez Seoane, professora de la Facultat en la qual s'emmarca aquest projecte, és encarregada de dirigir i guiar a l'autor durant tot el transcurs del projecte. Gràcies a aquest rol, l'estudiant s'assegura un desenvolupament correcte de la documentació i del sistema.

2.3.4 Protectores d'animals

Aquest actor desenvolupa un paper molt important per assegurar el correcte funcionament de l'aplicació. La seva aportació es basarà a facilitar-nos dades dels animals i dades de contacte per poder anunciar els seus animals en la nostra plataforma. A canvi, les protectores es beneficiaran de l'augment d'audiència que PetsWorld els aportarà.

2.3.5 Professionals relacionats amb els serveis animal

Les empreses relacionades amb els serveis animals, com veterinaris o passejador de gossos, podran anunciar els seus serveis a través de la plataforma. Aquests actors aportaran riquesa al contingut de l'aplicació i augmentarà el tràfic d'usuaris a la plataforma, a la vegada que aquestes empreses podran beneficiar-se de la publicitat.

2.3.6 Professionals relacionats amb els productes animals

Per aportar encara més valor a l'aplicació, les empreses relacionades amb els productes animals també podran anunciar els seus productes a través de la plataforma. Aquest grup juga un paper idèntic a les empreses relacionades amb els serveis animals.

2.3.7 Particulars

Encara que les gosseres i les empreses relacionades amb el món animal també son usuaris, els usuaris finals als quals la plataforma va dirigida, són els particulars; que participaran comprant o adoptant dins l'aplicació. Aquests usuaris finals es beneficiaran dels serveis i facilitats que oferirà la plataforma, i a la vegada aportaran valors econòmics i socials.

2.3.8 Mascotes

Segurament els actors més beneficiats d'aquest projecte són les mascotes. Gràcies a PetsWorld les mascotes tindran una plataforma dedicada a elles, on cuidar per al seu benestar és un dels principals pilars de l'aplicació.

3 Abast

3.1 Objectius genèrics

Basats en els problemes destacats anteriorment (secció 2.1), existeixen dos objectius principals en aquest treball: crear una aplicació per a mòbil, amb el corresponent servidor; i explotar les facilitats que aporta el Framework Flutter, per així millorar notòriament l'UI i la UX d'aquesta app. Seguidament s'explicarà més detalladament cada un d'aquests objectius. A més, encara que en aquest treball de final de grau només es desenvoluparà l'aplicació per a mòbil, és important recordar que aquest treball forma part d'un projecte més gran, on el resultat final serà una plataforma única per a web i mòbil.

La finalitat de desenvolupar un portal d'anuncis d'adopcions de mascotes és donar màxima accessibilitat a l'adopció per internet. Per tal d'assegurar el compliment d'aquest objectiu i així causar un impacte positiu en el mercat, aquesta aplicació necessitarà distingir-se de les plataformes existents. Així mateix, per tal d'incrementar el flux del sistema i abastar més usuaris, aquesta plataforma inclourà diverses seccions relacionades amb el món de les mascotes, tal com a articles i serveis. A més, l'aplicació serà multi-llenguatge i compatible amb Android i iOS.

D'altra banda, a part d'oferir-nos la possibilitat de crear una aplicació nativa multiplataforma des d'un sol codi font, Flutter també aporta facilitat en implementar dissenys actuals o innovadors, seguint normatives de disseny com Material Design⁶ [9]. Addicionalment, aquest framework proporciona eines per crear tests automàtics. Gràcies a això es podrà realitzar proves per assegurar el correcte funcionament de l'aplicació durant el desenvolupament d'aquesta.

3.2 Objectius específics

Per tal d'assolir correctament els objectius genèrics, s'hauran d'assolir els següents objectius específics.

3.2.1 Objectius tècnics

Els objectius tècnics estan directament relacionats amb les eines que es faran servir per a cada bloc. Així mateix, l'objectiu específic tècnic relacionat amb el **front-end** és el següent:

- Familiaritzar-se amb les eines de testing, facilitades per Flutter.

Atès que l'autor del treball ja està familiaritzat amb el llenguatge Dart i el framework Flutter, només caldrà familiaritzar-se amb les eines de testing esmentades anteriorment.

⁶ Normativa de disseny enfocada al sistema Android. L'objectiu d'aquestes és assegurar una correcta implementació d'UX.

Quant al back-end, es farà ús del framework Express [10], juntament amb diverses eines com NodeJS [11], MongoDB [12], i GraphQL [13]. Llavors, els objectius específics tècnics relacionats amb el **back-end** són:

- Familiaritzar-se amb el framework Express.
- Familiaritzar-se amb el llenguatge JavaScript⁷.
- Familiaritzar-se amb l'entorn d'execució NodeJS.
- Familiaritzar-se amb bases de dades no relacionals, i més concretament amb MongoDB.
- Familiaritzar-se amb GraphQL.

3.2.2 Objectius funcionals

Els objectius específics funcionals són els següents:

- Permetre a l'usuari canviar l'idioma entre català, castellà i anglès.
- Permetre a l'usuari registrar-se, iniciar sessió, i tancar sessió en el sistema.
- Permetre a l'usuari modificar les seves dades personals.
- Permetre a protectores i particulars crear i eliminar anuncis sobre animals en adopció.
- Permetre a professionals crear i eliminar anuncis sobre articles.
- Permetre a professionals, i particulars crear i eliminar anuncis sobre serveis.
- Mostrar a l'usuari els animals disponibles per adoptar de la base de dades i totes les dades relacionades a aquests disponibles.
- Permetre als usuaris filtrar protectores per proximitat.
- Implementar un sistema de puntuació, per permetre als usuaris valorar altres perfils.
- Implementar tests unitaris⁸ que assegurin el correcte funcionament del codi.

3.3 Obstacles i riscos

Finalment, en tot desenvolupament existeixen diversos obstacles i riscos que poden frenar o endarrerir el curs del projecte. A continuació es llista els obstacles i riscos més probables:

3.3.1 Limitació temporal en l'entrega

Tenir una data fixe per a l'entrega final del producte és un risc que pot arribar a forçar al desenvolupador a prendre decisions dràstiques envers al desenvolupament.

⁷ JavaScript és el llenguatge en el que el Framework Express està implementat.

⁸ Un test unitari valida el correcte funcionament d'una sola funció, mètode o classe.

3.3.2 Inexperiència de les tecnologies

El fet que el desenvolupador no tingui experiència en certes eines del back-end que es faran servir en el desenvolupament pot ser un gran risc a l'hora de planificar el temps i els esforços de la implementació. Això pot provocar una gran desviació en el desenvolupament real.

3.3.3 Inexistència de llibreries

Com s'ha explicat anteriorment, el front-end d'aquest treball es realitzarà amb el framework Flutter. Aquest entorn de treball és relativament nou, ja que la versió 1 va ser publicada oficialment el desembre del 2018. A conseqüència d'aquest fet, és possible que per implementar certes funcionalitats no hi hagi llibreries específiques, o que algunes funcionalitats no es puguin implementar directament amb Flutter.

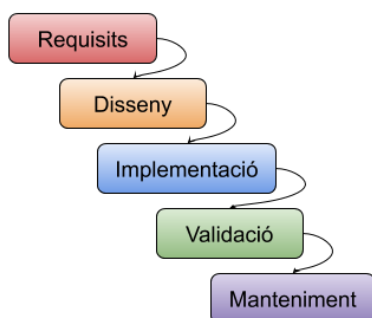
3.3.4 Coronavirus

La realització d'aquest treball, correspon en el marc temporal amb la pandèmia del virus COVID-19. Per tant, existeix la incertesa que el treball es pugui endarrerir, ja sigui perquè a escala individual l'autor del treball s'infecti, o perquè la institució en la qual s'ha emmarcat aquest treball paritzi els actes d'avaluació.

4 Metodologia i gestió del projecte

4.1 Mètode de treball

Antigament la metodologia que s'utilitzava per realitzar un projecte d'enginyeria de software era el model cascada.



IL·LUSTRACIÓ 2: MODEL CASCADA - ELABORACIÓ PRÒPIA

En la Il·lustració 2 es pot veure l'ordre seqüencial de les diferents fases que componen un desenvolupament de software.

Aquesta metodologia seqüencial és molt estricta, i això comporta certs avantatges i desavantatges respecte altres.

Un desavantatge important és que no es pot tornar a fases enrere. Això significa que si hi ha un problema de disseny, la fase d'implementació es pot complicar molt. A més, segons un informe publicat per *Standish Group* [14], una

metodologia *Agile* té 4 cops més possibilitats d'èxit que la metodologia en cascada.

Per aquests motius el desenvolupament d'aquest projecte es durà a terme fent ús d'una **metodologia Agile**. Aquesta metodologia es caracteritza per les següents qualitats: desenvolupament evolutiu i flexible, planificació i comunicació. Per aconseguir això, les metodologies *Agile* es basen en el manifest *Agile* [15].

Encara que existeixen diverses metodologies Agile, en aquest projecte es farà ús de la **metodologia Kanban** [16]. Aquesta metodologia simplifica la planificació i l'assignació de responsabilitats a través d'un taulell – representat en la Il·lustració 3-, on com a mínim, haurien d'existir tres columnes: *to do*, *in progress*, i *done*. *Kanban* és una metodologia de les més usades, junt amb *Scrum* [17]. La raó per la qual s'ha escollit aquesta és perquè en el desenvolupament d'aquest projecte només participa una persona a l'equip desenvolupador, i la metodologia *Scrum* està més orientada a equips, on participen diferents rols.

Tot i això, s'aprofitarà el concepte de **Sprints** de la **metodologia Scrum**. Aquest concepte tracta de fer iteracions contínues de desenvolupament incremental. Al principi del projecte es defineix un període de temps fixe per a cada iteració, en aquest projecte seran **dues setmanes per iteració**, i es dividiran els casos d'ús en els diferents *Sprints*.

En cada iteració es realitzarà una reunió retrospectiva per veure quines tasques han quedat pendents – les tasques pendents passen a la següent iteració -, i com millorar l'estratègia pel següent *Sprint*, regulant el factor tasques/Sprint.

En el taulell *Kanban* mencionat anteriorment, virtual o físic, s'utilitzen diferents targetes visuals on cada targeta representa una tasca. Dintre del taulell s'organitzen les tasques i es divideixen entre les columnes. Per un millor enteniment, vegeu la figura següent.



IL·LUSTRACIÓ 3: EXEMPLE D'UN TAULELL KANBAN [18]

En el marc d'aquest projecte, el taulell serà compost les 3 columnes mencionades anteriorment. Per cada història d'usuari s'assignaran unes tasques, que seran col·locades a la columna de "To do". La columna "In progress" contindrà les tasques que estan en desenvolupament actualment, i després de validar-les i passar els tests corresponents, es col·locaran les tasques a "Done", donant així per acabat el desenvolupament de la tasca corresponent.

4.1.1 Eines de seguiment

A continuació, en aquest apartat, es mostra una llista de les eines que s'utilitzaran durant el transcurs del desenvolupament de l'aplicació per validar que s'assoleixin els objectius correctament.

4.1.1.1 Git

Git [19] és un sistema de control de versions gratuït i de codi obert. S'utilitza per treballar en el desenvolupament de projectes de software cooperativament o dur a terme la gestió de versions i funcionalitats d'una manera pràctica i eficient. Un *git* que gestiona un projecte de software s'anomena *repositori*, i un repositori gestiona diferents versions d'un projecte.

4.1.1.2 GitKraken

GitKraken [20] és una, de les moltes, interfícies d'usuari per a Git. Aquesta aplicació destaca entre les altres per la seva UI intuïtiva, de manera que ajuda a entendre fàcilment el funcionament de Git.

4.1.1.3 Github

Github [21] és un gestor de repositoris git. Bàsicament el que Github ens ofereix és guardar els repositoris git al núvol, i així evitar la pèrdua del codi per qualsevol problema que pugui aparèixer posteriorment.

4.1.1.4 GitKraken Glo Boards

GitKraken Glo Boards [22] és una aplicació web que ens permet utilitzar taulers de Kanban virtuals amb les seves corresponents targetes. A més, com és una eina desenvolupada pel mateix equip que ha desenvolupat GitKraken, aquesta permet ser integrada dintre de la mateixa aplicació GitKraken. Així doncs, en poques paraules, aquesta eina facilitarà la implementació de la metodologia que hem descrit anteriorment.

4.2 Planificació inicial

4.2.1 Planificació temporal

L'objectiu d'aquest apartat és definir, amb la màxima precisió possible, les dates en què es desenvoluparà el projecte, el temps que s'hi dedicarà quantificant en hores i dies i, finalment, la data prevista d'entrega de la memòria.

Primerament, s'ha calculat que **el cost de la gestió del projecte és de 75h**, d'acord amb les especificacions de *GEP*.

En segon lloc, el període de desenvolupament del projecte comprèn les dades del 17/03/2020 al 30/07/2020, ambdues incloses. Aquesta etapa està composta per diferents fases:

- Anàlisi de requisits
- Especificació i disseny
- Implementació

En les primeres dues subetapes es realitzarà una jornada de 3h al dia, en canvi en la subetapa d'implementació es dedicarà una jornada de 6h. Aquesta subetapa serà dividida en 6 *Sprints* i dues iteracions preparatòries: inicial i final, que duren –la meitat d'un *Sprint* normal – 1 setmana. Per tant, s'ha estimat que **aquest període comporta 480 hores de treball**.

En tercer i últim lloc, s'ha definit que **la documentació de la memòria del TFG s'elabora paral·lelament** juntament amb el desenvolupament del projecte, comprnent així les dades del 17/03/2020 al 30/07/2020, ambdues incloses també. S'estima que la quantitat d'hores dedicades a la documentació són **50 hores**, sense tenir en compte la planificació ni la gestió del projecte inicial.

Per tant, **el total a dedicar en aquest projecte són 605 hores**, estimades.

4.2.2 Descripció de les tasques

Seguidament es mostren les tasques de cada etapa, mostrant així: un **codi identificador**, el **títol**, una **descripció** breu, **l'estimació d'hores** necessàries per a dur a terme la tasca, les **dependències de precedència**, i finalment els **recursos necessaris**.

4.2.2.1 Etapa: GEP

T001: Planificació i gestió del projecte

Descripció: Redacció i definició de la documentació inicial incloent entre d'altres: abast, objectius, planificació i pressupost.

Duració: 75h

Dependències: -

Recursos humans: Cap de projecte

Recursos materials: Ordinador

4.2.2.2 Etapa: Desenvolupament de l'aplicació

T003: Anàlisi de requisits

Descripció: Anàlisi de requisits per assegurar una correcta posterior implementació

Duració: 30h

Dependències: -

Recursos humans: Analista programador

Recursos materials: -

T004: Especificació i disseny

Descripció: -

Duració: 30h

Dependències: T003

Recursos humans: 60% programador sènior, 40% dissenyador UX

Recursos materials: Ordinador

T006: Iteració inicial

Descripció: En aquesta iteració s'estudiarà i s'implementarà la Mock API. També es prepararà el front-end per la següent iteració

Duració: 30h

Dependències: T004

Recursos humans: Programador júnior

Recursos materials: Ordinador

T007: Sprint 1

Descripció: Es dedicarà completament al front-end. S'implementarà el multilinguatge, la pantalla de registre, la d'inici de sessió, la de configuració de l'aplicació, i la de perfil i editar perfil; amb les seves corresponents lògiques

Duració: 60h

Dependències: T006

Recursos humans: Programador júnior

Recursos materials: Ordinador

T008: Sprint 2

Descripció: Seguint amb el front-end, s'implementarà la pantalla principal, junt amb les pantalles d'anuncis, la de crear anuncis i la d'editar anuncis; junt amb les seves corresponents lògiques

Duració: 60h

Dependències: T007

Recursos humans: Programador júnior

Recursos materials: Ordinador

T009: Sprint 3

Descripció: Seguint amb el front-end, s'implementarà el sistema de valoracions i puntuacions. Una pantalla per la cerca avançada d'anuncis, i la seva corresponent lògica, i el sistema de geolocalització.

Duració: 60h

Dependències: T008

Recursos humans: Programador júnior

Recursos materials: Ordinador

T010: Sprint 4**T011:** Sprint 5

Descripció: en aquest Sprint es farà refactor ⁹ de l'aplicació i altres retocs. A més, s'estudiaran les tecnologies del back-end i es prepararà per la següent iteració.	Descripció: Es començarà a implementar el back-end: CRUD ¹⁰ d'usuari, inici de sessions d'usuaris i CRUD d'anuncis.
Duració: 60h	Duració: 60h
Dependències: T008	Dependències: T009
Recursos humans: Programador júnior	Recursos humans: Programador júnior
Recursos materials: Ordinador	Recursos materials: Ordinador

T012: Sprint 6**T013:** Iteració final

Descripció: en aquest Sprint s'implementarà en el back-end la cerca avançada d'anuncis amb filtres, sistema de valoració, puntuació i geolocalització	Descripció: Es realitzarà refactor del back-end, es validarà que tots els requisits definits prèviament hagin sigut satisfets, i es deixarà tot preparat per la presentació
Duració: 60h	Duració: 30h
Dependències: T010	Dependències: T009
Recursos humans: Programador júnior	Recursos humans: Programador júnior
Recursos materials: Ordinador	Recursos materials: Ordinador

4.2.2.3 Etapa: Documentació i defensa

T014: Documentació i defensa

Descripció: Acabar la memòria i l'estudi de la presentació oral, per la defensa del treball.

Duració: 50h

Dependències: -

Recursos humans: Cap del projecte

Recursos materials: Ordinador

4.2.3 Diagrama de Gantt

Es pot visualitzar el diagrama de Gantt en el Annex 1, pàgina 102. A continuació, en la Taula 2 es pot veure un resum més visual del diagrama, amb els colors corresponents del diagrama, les seves dependències, i les hores al dia que es dedicaran en cada una.

⁹ Tècnica d'enginyeria de software per a reestructurar el codi font.

¹⁰ Sigles per crear, llegir, actualitzar i eliminar en anglès; operacions bàsiques de gestió de dades.

Codi	Tasca	Hores estimades	Dependències	Data inici	Data final
T001	Planificació i gestió del projecte	75		17/02	16/03
T002	Desenvolupament de l'aplicació	480		17/03	30/07
T003	Anàlisi de requisits	30	T001	17/03	31/03
T004	Especificació i disseny	30	T003	1/04	15/04
T005	Implementació	420		16/04	30/07
T006	Iteració inicial	30	T004	16/04	23/04
T007	Sprint 1	60	T006	24/04	8/05
T008	Sprint 2	60	T007	9/05	23/05
T009	Sprint 3	60	T008	24/05	7/06
T010	Sprint 4	60	T009	8/06	22/06
T011	Sprint 5	60	T010	23/06	7/07
T012	Sprint 6	60	T011	8/07	22/07
T013	Iteració final	30	T012	23/07	30/07
T014	Documentació i defensa	50		17/03	16/10
T015	Redacció de la memòria i reunions de seguiment	40			
T016	Preparació i presentació oral	10			
TOTAL		605		17/02	23/10

TAULA 2: ETAPES DEL DESENVOLUPAMENT

4.2.4 Gestió del risc: obstacles i plans de mitigació

Anteriorment s'han mencionat els possibles obstacles i riscos que comporta aquest projecte (pàg. 16). En la següent taula, Taula 3, s'ha intentat mesurar la probabilitat, l'impacte, i quin pla de mitigació s'aplicarà en cas de ser necessari.

Risc	Probabilitat	Impacte	Pla de mitigació
Limitació temporal en l'entrega	Alta	Baix	El pla de mitigació és el mateix per els tres riscos mencionats: S'ha sobreestimat les tasques més complexes tenint en compte que poden requerir un procés d'aprenentatge inicial. A més, al planificar les iteracions s'ha tingut en compte aquests riscos, planificant així menys pes de tasques en l' <i>Sprint</i> 4 i la iteració final, resultant així amb temps extra en aquestes iteracions.
Inexperiència de les tecnologies	Alta	Alt	
Coronavirus	Alta	Mitjà	

TAULA 3: GESTIÓ DELS RISCOS

4.3 Gestió econòmica

4.3.1 Identificació i estimació dels costos

4.3.1.1 Costos directes per activitat

En la Taula 4 es pot veure els diferents rols que formaran part del projecte, la seva remuneració horària i el seu cost total basat en la dedicació de cada rol. Aquesta remuneració econòmica s'ha obtingut a partir de la plataforma online per buscar feina *Indeed* [23]. S'ha extret el sou net mig per any, i seguidament s'ha calculat el preu per hora considerant 4 setmanes al mes de 40 hores laborables per setmana.

Rol	Remuneració (en €/h)	Hores totals	Cost total (en €)
Cap de projecte	37,00	125	4.625,00
Analista programador	15,70	30	471,00
Programador sènior	16,38	18	294,75
Programador júnior	9,00	420	3.780,00
Dissenyador gràfic	9,38	12	112,50
TOTAL			9.283,25 €

TAULA 4: COSTOS DIRECTES PER ACTIVITAT

Per aclarir, les hores totals de la taula anterior corresponen a les hores totals dedicades per cada rol, segons s'ha definit prèviament a l'apartat on s'han definit les tasques.

4.3.1.2 Costos indirectes

Aquest projecte es durà a terme entre una casa particular i les instal·lacions de la UPC, per tant no es tenen en compte les despeses relacionades amb la llum, aigua, internet, etc.

- **Transport:** Entre la casa particular i les instal·lacions de la UPC hi ha 3,5km de distància. Aquest recorregut es fa amb una bicicleta ja amortitzada, per tant el cost del transport és nul.
- **Hardware:** L'ordinador que requereix aquest projecte és un Macbook Pro, ja que per desenvolupar i testejar una aplicació mòbil per a iOS es necessita un ordinador de la mateixa companyia. Aquest ordinador ja està amortitzat, així que el cost del hardware també és nul
- **Software:** No és necessari comprar cap software per desenvolupar aquest projecte, per tant el cost del software també és nul.

Resumint, les **despeses totals derivades de factors indirectes són 0€.**

4.3.1.3 Contingències

En el cas d'aquest projecte, s'ha decidit reservar un 20% del pressupost per la partida de contingències, és a dir, com a sobre cost genèric per cobrir obstacles no previstos. En la taula que hi ha a continuació es pot veure el càlcul resultant:

	Preu	Percentatge	Cost contingència
Costos directes	9.283,25 €	20 %	1.856,65 €
Costos indirectes	0,00 €	20 %	0,00 €
TOTAL			1.856,65 €

TAULA 5: CONTINGÈNCIES

4.3.1.4 Imprevistos

El principal imprevist que pot sorgir al projecte és una significativa desviació del temps deguda als obstacles definits anteriorment. Es calcula que té un 80% de possibilitat que passi, i que podria suposar un augment d'un 20% del temps estimat inicialment. Per tant, **el cost dels imprevistos és 1.485,32 €.** Aquest valor s'ha calculat amb la següent equació: $9.283,25 \text{ €} * 0,8 * 0,2$, sent 9.283,25 € el cost total dels recursos humans.

4.3.1.5 Cost total

En la següent taula es mostra el sumatori de tots els costos, formant així el pressupost final.

Costos directes	Costos indirectes	Contingències	Imprevistos	Total
9.283,25 €	0 €	1.856,65 €	1.485,32 €	12.625,22 €

TAULA 6: PRESSUPOST FINAL

4.3.2 Control de gestió

Els recursos materials són una inversió inicial, i per tant, només es podrà comparar el cost estimat amb el cost real un cop realitzada la inversió.

Pel que fa als recursos humans, en canvi, sí que s'ha de realitzar un control constant durant tot el projecte. Concretament s'analitzaran les despeses al final de cada una de les fases del projecte i es compararan amb les estimacions realitzades inicialment. A partir d'aquestes dades se'n calcularà les desviacions. D'aquesta manera si en algun moment es produeix una desviació total notable es procedirà a intentar corregir l'estimació del pressupost.

Si totes aquestes desviacions potencials produeixen que el pressupost inicial no és suficient, s'utilitzarà el fons de contingències per a cobrir aquestes despeses afegides.

Per calcular les desviacions s'utilitzarà les següents fórmules:

- Desviació de la mà d'obra en preu = (cost estimat – cost real) * consum d'hores real.
- Desviació a la realització d'una tasca en preu = (cost estimat – cost real) * consum d'hores real.
- Desviació en el total de la realització de les tasques = cost total tasques estimat – cost total tasques reals.
- Desviació en el total dels recursos = cost total recursos estimats – cost total recursos reals.
- Desviació en el total dels costos fixes = costos fixes totals estimats – costos fixes totals reals.

5 Anàlisi de requisits i especificació

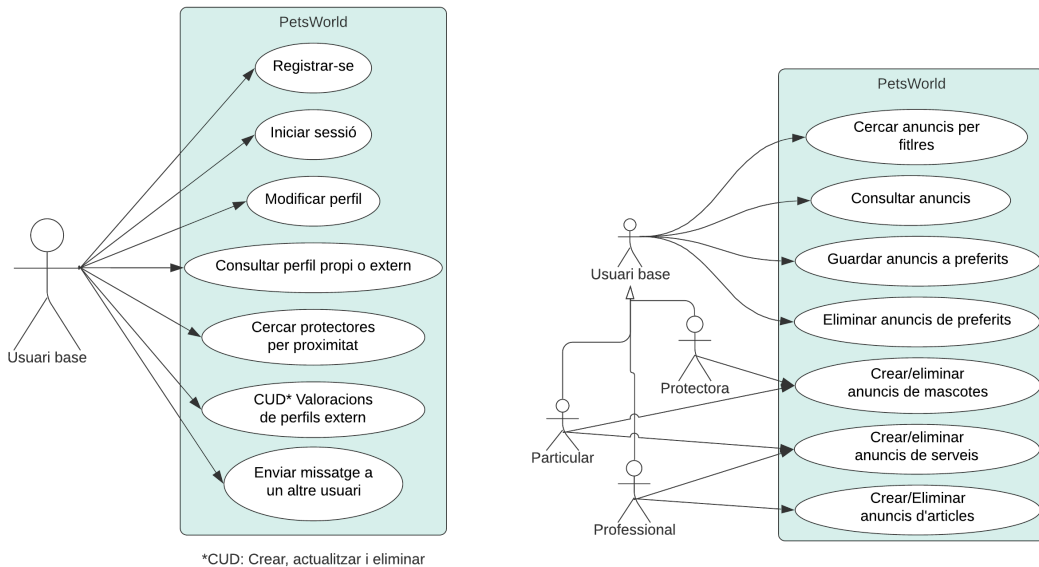
Segons l'estàndard IEEE [24], un requisit és una condició o capacitat del sistema que un usuari necessita per poder resoldre un problema o assolir un objectiu. A continuació es descriuen els requisits del sistema, tant funcionals com no funcionals, que s'han tingut en compte per a assegurar un resultat de qualitat. Donat que l'aplicació no serà desenvolupada per a un client real, els requisits següents han sigut generats per l'autor del projecte.

5.1 Requisits funcionals

En aquest apartat es defineixen els requisits funcionals mitjançant un diagrama genèric de casos d'ús relacionats amb els actors principals del sistema (actors implicats definits en la secció 2.3).

5.1.1 Diagrama genèric de casos d'ús

Seguidament s'exposa el diagrama de tots els casos d'ús del sistema:



IL·LUSTRACIÓ 4: DIAGRAMA GENÈRIC DE CASOS D'ÚS – ELABORACIÓ PRÒPIA

5.1.2 Descripció dels casos d'ús

Es descriuran cada un dels casos d'ús mitjançant una taula on es detalla:

- **Actor principal:** actor associat al cas d'ús.
- **Precondició:** totes aquelles condicions que s'han de complir en el moment en el qual el cas d'ús s'executi.
- **Disparador:** acció que activa el cas d'ús.
- **Escenari principal d'èxit:** llistat d'accions que succeeixen en el cas que tot funcioni correctament.
- **Extensions:** seguit d'accions que se succeeixen en el cas que alguna acció de l'escenari principal no hagi funcionat correctament.

5.1.2.1 R001 – Registrar-se

Actor principal
Usuari base
Disparador
L'actor vol registrar-se en l'aplicació
Precondició
-

Escenari principal d'èxit
<ol style="list-style-type: none"> 1. L'actor omple el formulari de la pantalla de registrar-se 2. L'actor fa click en el botó de registrar-se 3. El sistema valida les dades i són correctes 4. El sistema autoritza a l'usuari per a utilitzar l'aplicació
Extensions
<ol style="list-style-type: none"> 4.a. El sistema valida les dades i no són correctes 4.b. El sistema mostra per pantalla un missatge de text indicant que les dades no són correctes <p><i>Es repeteix el pas 3 fins que l'usuari decideix-hi acabar</i></p>

TAULA 7: DESCRIPCIÓ DEL CAS D'ÚS "REGISTRAR-SE"

5.1.2.2 R002 – Iniciar sessió

Actor principal
Usuari base
Disparador
L'actor vol iniciar sessió en l'aplicació
Precondició
L'usuari s'ha registrat prèviament
Escenari principal d'èxit
<ol style="list-style-type: none"> 1. L'actor omple el formulari de la pantalla d'inici de sessió 2. L'actor fa click en el botó d'iniciar sessió 3. El sistema valida les dades i són correctes 4. El sistema autoritza a l'usuari per a utilitzar l'aplicació

TAULA 8: DESCRIPCIÓ DEL CAS D'ÚS "INICIAR SESSIÓ"

5.1.2.3 R003 – Modificar perfil

Actor principal
Usuari base
Disparador
L'actor vol modificar les dades del seu perfil
Precondició
L'usuari s'ha registrat prèviament i ha iniciat sessió en l'aplicació
Escenari principal d'èxit

1. L'actor es dirigeix a la pantalla d'editar el seu perfil
2. L'actor modifica les dades que vol modificar
3. El sistema valida les dades i són correctes
4. El sistema modifica les dades del perfil d'aquest actor

TAULA 9: DESCRIPCIÓ DEL CAS D'ÚS "MODIFICAR PERFIL"

5.1.2.4 R004 – Consultar perfil propi o extern

Actor principal
Usuari base
Disparador
L'actor vol visualitzar el perfil d'un altre usuari, o el seu mateix
Precondició
L'usuari s'ha registrat prèviament i ha iniciat sessió en l'aplicació
Escenari principal d'èxit
<ol style="list-style-type: none"> 1. L'actor es dirigeix a la pantalla de visualitzar perfils externs o el propi 2. El sistema mostra les dades disponibles a l'usuari

TAULA 10: DESCRIPCIÓ DEL CAS D'ÚS "CONSULTAR PERFIL PROPI O EXTERN"

5.1.2.5 R005 – Enviar missatge a un altre usuari

Actor principal
Usuari base
Disparador
L'actor vol enviar un missatge a un altre usuari
Precondició
<ol style="list-style-type: none"> 1. L'usuari s'ha registrat prèviament i ha iniciat sessió en l'aplicació. 2a. Hi ha anuncis en el sistema. 2b. Hi ha una conversació iniciada amb l'usuari en qüestió.
Escenari principal d'èxit
<ol style="list-style-type: none"> 1a. L'actor es dirigeix a la pantalla de visualitzar anuncis 2a. L'actor procedeix a veure la informació detallada d'un anunci que no és seu. 3a. L'actor fa clic en el botó de contactar 4a. L'actor escriu un missatge 1b. L'actor es dirigeix a la pantalla de xats. 2b. L'actor entra al xat corresponent. 3b. L'actor escriu un missatge i l'envia.

TAULA 11: DESCRIPCIÓ DEL CAS D'ÚS "ENVIAR UN MISSATGE A UN ALTRE USUARI"

5.1.2.6 R006 – Consultar anuncis

Actor principal
Usuari base
Disparador
L'actor vol consultar anuncis
Precondició
<ol style="list-style-type: none"> 1. Hi ha anuncis publicats en el sistema
Escenari principal d'èxit
<ol style="list-style-type: none"> 1. L'actor es dirigeix a la pantalla inicial de l'aplicació 2. El sistema mostra els anuncis

TAULA 12: DESCRIPCIÓ DEL CAS D'ÚS "CONSULTAR ANUNCIS"

5.1.2.7 R007 – Cercar anuncis per filtres

Actor principal
Usuari base
Disparador
L'actor vol cercar anuncis per filtres específics
Precondició
1. Hi ha anuncis publicats en el sistema amb els filtres seleccionats
Escenari principal d'èxit
<ol style="list-style-type: none"> 1. L'actor es dirigeix a la pantalla de cerca avançada d'anuncis 2. El sistema mostra els diferents camps disponibles per afegir com a filtre en la cerca 3. L'actor selecciona els filtres desitjats i fa click al botó de cerca 4. El sistema mostra els anuncis que compleixen els filtres especificats per l'actor

TAULA 13: DESCRIPCIÓ DEL CAS D'ÚS "CERCA ANUNCIS PER FILTRES"

5.1.2.8 R008 – Cercar protectores per proximitat

Actor principal
Usuari base
Disparador
L'actor vol cercar protectores per proximitat
Precondició
1. Hi ha protectores en el sistema
Escenari principal d'èxit
<ol style="list-style-type: none"> 1. L'actor es dirigeix a la pantalla de categories 2. El sistema mostra les diferents categories d'anuncis 3. L'actor selecciona la categoria "Protectores" 4. El sistema mostra els perfils de protectores ordenats per proximitat

TAULA 14: DESCRIPCIÓ DEL CAS D'ÚS "CERCA PROTECTORES PER PROXIMITAT "

5.1.2.9 R009 – Crear, actualitzar o eliminar una valoració a un perfil extern

Actor principal
Usuari base
Disparador

L'actor vol crear, actualitzar o eliminar una valoració a un perfil extern
Precondició
<ol style="list-style-type: none"> 1. Existeixen altres usuaris a part de l'usuari base 2a. - 2b. Existeix una valoració prèvia per editar-la 2c. Existeix una valoració prèvia per eliminar-la
Escenari principal d'èxit
<ol style="list-style-type: none"> 1. L'actor es dirigeix a la pantalla de perfil de l'usuari que vol crear, actualitzar o eliminar la valoració 2. El sistema mostra les diferents opcions <ol style="list-style-type: none"> 3a. L'actor selecciona l'opció "Valorar perfil" 4a. El sistema deixa a l'usuari introduir un comentari, junt amb una puntuació 5a. L'actor omple les dades necessàries 6a. El sistema processa i mostra la nova valoració de l'actor en el perfil extern 3b. L'actor selecciona l'opció editar valoració <ol style="list-style-type: none"> 4b. El sistema deixa a l'usuari editar la seva valoració 5b. L'actor edita les dades 6b. El sistema processa i actualitza la nova valoració 3c. L'actor selecciona l'opció eliminar valoració <ol style="list-style-type: none"> 4c. El sistema processa i elimina la valoració corresponent

TAULA 15: DESCRIPCIÓ DEL CAS D'ÚS "VALORAR PERFIL EXTERN "

5.1.2.10 R010 – Crear o eliminar anuncis de mascotes

Actor principal
Protectora o particular
Disparador
L'actor vol publicar o eliminar un anunci de mascota
Precondició
<ol style="list-style-type: none"> 1. L'usuari s'ha registrat prèviament i ha iniciat sessió en l'aplicació 2. Si l'acció és eliminar, prèviament ha d'existir l'anunci de mascota creat per el mateix usuari

Escenari principal d'èxit

- 1a. L'actor vol crear un anunci, i es dirigeix a la pantalla de pujar anunci en l'apartat de mascotes
- 2a. El sistema crea l'anunci

- 1b. L'actor vol eliminar un anunci, i es dirigeix a la pantalla de l'anunci que vol eliminar
- 2b. L'actor fa click al botó d'eliminar anunci
- 3b. El sistema elimina l'anunci de la base de dades

TAULA 16: DESCRIPCIÓ DEL CAS D'ÚS "CREAR O ELIMINAR ANUNCIS DE MASCOTES"

5.1.2.11 R011 – Crear o eliminar anuncis de serveis

Actor principal

Professional o particular

Disparador

L'actor vol publicar o eliminar un anunci de servei relacionat amb mascotes

Precondició

- 1. L'usuari s'ha registrat prèviament i ha iniciat sessió en l'aplicació
- 2. Si l'acció és eliminar, prèviament ha d'existir l'anunci de servei creat per el mateix usuari

Escenari principal d'èxit

- 1a. L'actor vol crear un anunci, i es dirigeix a la pantalla de pujar anunci en l'apartat de serveis
- 2a. El sistema crea l'anunci

- 1b. L'actor vol eliminar un anunci, i es dirigeix a la pantalla de l'anunci que vol eliminar
- 2b. L'actor fa click al botó d'eliminar anunci
- 3b. El sistema elimina l'anunci de la base de dades

TAULA 17: DESCRIPCIÓ DEL CAS D'ÚS "CREAR O ELIMINAR ANUNCIS DE SERVEIS"

5.1.2.12 R012 – Crear o eliminar anuncis d'articles

Actor principal

Professional

Disparador

L'actor vol publicar o eliminar un anunci d'un article relacionat amb mascotes

Precondició

- 1. L'usuari s'ha registrat prèviament i ha iniciat sessió en l'aplicació

2. Si l'acció és eliminar, prèviament ha d'existir l'anunci de l'article creat per el mateix usuari

Escenari principal d'èxit

1a. L'actor vol crear un anunci, i es dirigeix a la pantalla de pujar anunci en l'apartat d'articles

2a. El sistema crea l'anunci

1b. L'actor vol eliminar un anunci, i es dirigeix a la pantalla de l'anunci que vol eliminar

2b. L'actor fa click al botó d'eliminar anunci

3b. El sistema elimina l'anunci de la base de dades

TAULA 18: DESCRIPCIÓ DEL CAS D'ÚS "CREAR O ELIMINAR ANUNCIS D'ARTICLES"

5.1.2.13 R013 – Guardar anuncis a preferits

Actor principal

Usuari base

Disparador

L'actor vol guardar anuncis com a preferits, per la seva posterior localització

Precondició

1. L'usuari s'ha registrat prèviament i ha iniciat sessió en l'aplicació

2. El sistema ha de disposar d'anuncis

Escenari principal d'èxit

1. L'actor es dirigeix a la pantalla d'un anunci concret

2. L'actor fa click al botó corresponent per a guardar-lo

3. El sistema afegeix una referència de l'anunci en la secció "Preferits" de l'usuari

TAULA 19: DESCRIPCIÓ DEL CAS D'ÚS "GUARDAR ANUNCIS A PREFERITS"

5.1.2.14 R014 – Eliminar anuncis de preferits

Actor principal

Usuari base

Disparador

L'actor vol eliminar anuncis de preferits.

Precondició

1. L'usuari s'ha registrat prèviament i ha iniciat sessió en l'aplicació

2. El sistema ha de disposar d'anuncis

3. L'usuari ha de tenir anuncis com a preferits.

Escenari principal d'èxit

1. L'actor es dirigeix a la pantalla de preferits.

2. L'actor es dirigeix a la informació detallada dels anuncis els quals vol eliminar de preferits

3. L'actor clica el botó corresponent de preferits per eliminar-lo.

4. El servei elimina la referència de l'anunci eliminat de la secció "Preferits" de l'usuari.

TAULA 20: DESCRIPCIÓ DEL CAS D'ÚS "ELIMINAR ANUNCIS DE PREFERITS"

5.2 Requisits no funcionals

Seguidament es mostraran els requisits no funcionals definits per l'autor, però prèviament s'explicarà què indica cada apartat de les taules, que es basen en les plantilles de VOLERE¹¹.

- **Número de requisit:** és un identificador, per poder referenciar-lo més endavant.
- **Descripció:** el requisit en si, el que volem que es compleixi en el sistema.
- **Justificació:** quin és el motiu pel qual es vol que es compleixi.
- **Criteri de satisfacció:** com se sap si el requisit s'ha complert.
- **Satisfacció de l'usuari:** de l'1 al 5 (de menys a més), com de satisfet se sentirà l'usuari si es compleix el requisit.
- **Insatisfacció de l'usuari:** de l'1 al 5 (de menys a més), com d'insatisfet se sentirà l'usuari si no es compleix el requisit.
- **Prioritat:** la importància que té en el sistema que aquest requisit es compleixi o no.

5.2.1.1 Requisits de percepció

Número de requisit	R015	Prioritat	Alta
Descripció	La vista ofereix diferents temes per diferents gustos visuals.		
Justificació	La interfície ha de resultar còmode a l'usuari, proporcionant així una aparença més adaptada a aquest.		
Criteri de satisfacció	La vista proporciona un tema clar i un tema fosc, i deixa a l'usuari decidir quin tema aplicar.		
Satisfacció de l'usuari	4	Insatisfacció de l'usuari	4

TAULA 21: TAULA DE VOLERE DEL REQUISIT R015

¹¹ VOLERE és una metodologia que treballa per a l'adquisició i anàlisi de requisits

Número de requisit	R016	Prioritat	Alta
Descripció	El disseny de la interfície resulta atractiu per a l'usuari		
Justificació	La interfície resultarà agradable a la vista dels usuaris, per així facilitar les interaccions en l'aplicació		
Criteri de satisfacció	Es realitzarà una enquesta als usuaris que hagin utilitzat el sistema més d'un cop, i el 80% o més confirmarà la satisfacció del requisit		
Satisfacció de l'usuari	4	Insatisfacció de l'usuari	4

TAULA 22: TAULA DE VOLERE DEL REQUISIT R016

Número de requisit	R017	Prioritat	Alta
Descripció	El disseny de la interfície resulta senzill d'utilitzar per a l'usuari		
Justificació	Els usuaris poden no tenir un alt nivell en aplicacions mòbils, per això la interfície haurà de ser simple		
Criteri de satisfacció	Es realitzarà una enquesta als usuaris que hagin utilitzat el sistema més d'un cop, i el 80% o més confirmarà la satisfacció del requisit		
Satisfacció de l'usuari	5	Insatisfacció de l'usuari	5

TAULA 23: TAULA DE VOLERE DEL REQUISIT R017

5.2.1.2 Requisits de capacitat d'ús i d'humanitat

Número de requisit	R018	Prioritat	Alta
Descripció	El sistema té un llenguatge comprensible per a un usuari normal		
Justificació	Els usuaris han d'entendre el llenguatge del sistema fàcilment		
Criteri de satisfacció	S'utilitzarà un llenguatge clar i concret. A més, es proporcionarà l'habilitat de canviar l'idioma de l'aplicació. Durant l'enquesta als usuaris, es valorarà que aquests entenguin bé el contingut de l'aplicació.		
Satisfacció de l'usuari	5	Insatisfacció de l'usuari	5

TAULA 24: TAULA DE VOLERE DEL REQUISIT R018

Número de requisit	R019	Prioritat	Mitja
Descripció	El sistema utilitza icones fàcils d'identificar		
Justificació	Millorar al màxim la comprensió de l'usuari		

Criteri de satisfacció	S'intentarà, quan sigui possible, utilitzar icones ja establerts pel framework Flutter, adherits al Material Design		
Satisfacció de l'usuari	3	Insatisfacció de l'usuari	4

TAULA 25: TAULA DE VOLERE DEL REQUISIT R019

Número de requisit	R020	Prioritat	Alta
Descripció	Els textos que mostra l'aplicació són totalment llegibles		
Justificació	Per a interaccionar amb l'usuari, l'aplicació ha de assegurar que els textos siguin llegibles		
Criteri de satisfacció	Es seguirà les pautes establertes per Material Design per assegurar un correcte ús dels colors per a textos		
Satisfacció de l'usuari	1	Insatisfacció de l'usuari	5

TAULA 26: TAULA DE VOLERE DEL REQUISIT R020

5.2.1.3 Requisits d'acompliment

Número de requisit	R021	Prioritat	Mitja
Descripció	El sistema és escalable		
Justificació	El sistema ha d'estar preparat per l'augment d'usuaris		
Criteri de satisfacció	S'utilitzen eines, com MongoDB, tenint en compte l'escalabilitat d'aquestes		
Satisfacció de l'usuari	5	Insatisfacció de l'usuari	3

TAULA 27: TAULA DE VOLERE DEL REQUISIT R021

Número de requisit	R022	Prioritat	Mitja
Descripció	El sistema guarda les preferències de l'usuari encara que s'hagi sortit de l'aplicació		
Justificació	El sistema ha de recordar les preferències de l'usuari.		
Criteri de satisfacció	Es guardaran les dades de les preferències en local, i al entrar de nou a l'aplicació, el sistema recuperarà aquestes dades.		
Satisfacció de l'usuari	3	Insatisfacció de l'usuari	4

TAULA 28: TAULA DE VOLERE DEL REQUISIT R022

5.2.1.4 Requisits de seguretat

Número de requisit	R023	Prioritat	Alta
Descripció	El sistema ha de guardar la informació sensible dels usuaris de forma encriptada o hashejada		
Justificació	El sistema ha de protegir la informació de l'usuari contra atacs virtuals		
Criteri de satisfacció	S'encriptarà o es hashejarà tota informació sensible		
Satisfacció de l'usuari	1	Insatisfacció de l'usuari	5

TAULA 29: TAULA DE VOLERE DEL REQUISIT R023

5.2.1.5 Requisits legals

Número de requisit	R024	Prioritat	Alta
Descripció	Totes les dades de caràcter personal han de ser tractades complint la LOPD Espanyola (Llei orgànica de protecció de dades 15/1999)		
Justificació	El sistema ha de complir amb la legislació nacional		
Criteri de satisfacció	S'encriptarà o es hashejarà tota informació sensible		
Satisfacció de l'usuari	2	Insatisfacció de l'usuari	5

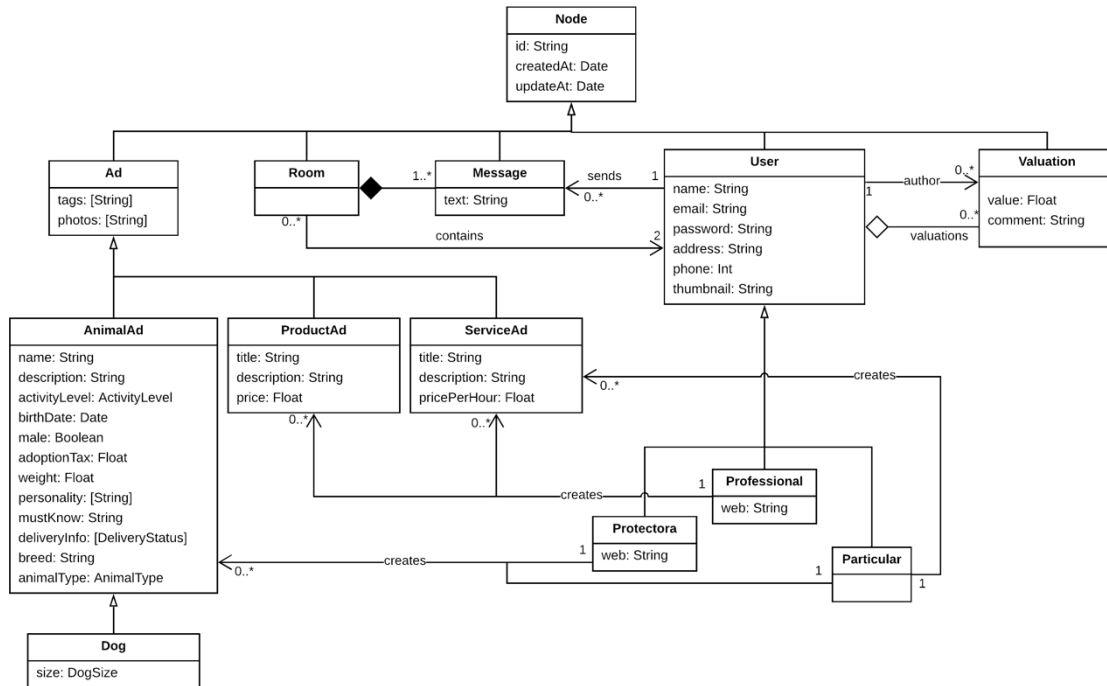
TAULA 30: TAULA DE VOLERE DEL REQUISIT R024

5.3 Model conceptual

Un model conceptual és la representació dels objectes significatius en el domini del problema. D'aquesta manera es mostra de forma més clara les classes d'objectes, els atributs de cada classe, les associacions entre elles i les restriccions d'integritat gràfiques i textuals. Convé ressaltar que un objecte no és més que una entitat que existeix al món real. A més té identitat pròpia sent així distingible dels altres objectes.

5.3.1 Esquema conceptual de les dades

En la següent il·lustració s'exposen les entitats del sistema, sent les més importants els anuncis i els usuaris, i les seves associacions.



IL·LUSTRACIÓ 5: ESQUEMA CONCEPTUAL DE DADES DEL SISTEMA

5.3.2 Restriccions d'integritat

Les restriccions d'integritat són condicions que o bé no es poden representar a l'esquema conceptual, o bé s'han omert per reduir la complexitat d'aquest. Aquestes restriccions són essencials perquè el sistema funcioni correctament, tal com s'ha dissenyat.

- Claus externes: (Node, id).
- Dos usuaris no poden tenir el mateix email.
- Un objecte Dog tindrà l'atribut animalType igual a Dog.
- Una Room no pot tenir dos usuaris amb el mateix id.
- El sender d'un Message ha de ser un usuari de la Room a la qual pertany el Message.
- El value d'una Valuation ha de ser més gran o igual que 0.
- El value d'una Valuation ha de ser més petit o igual a 5.
- Una Valuation no pot pertànyer a l'usuari *author*.

5.3.3 Descripció de les classes

5.3.3.1 Node

Representa una classe base per cada objecte derivat el qual s'identificarà per l'atribut id.

- **id:** Identificador de l'objecte.
- **createdAt:** Data i moment exacte en el qual s'ha creat l'objecte.

- **updatedAt:** Data i moment exacte de l'última modificació de l'objecte.

5.3.3.2 User

Interfície que representa els usuaris registrats que interaccionen amb el sistema. Les úniques classes que poden implementar aquesta interfície són els Particulars, Protectores, i Professionals.

- **name:** Nom de l'usuari.
- **email:** Correu electrònic de l'usuari.
- **password:** Contrasenya de l'usuari per accedir al sistema.
- **address:** Adreça física de l'usuari.
- **phone:** Número de telèfon o mòbil de l'usuari.
- **thumbnail:** Adreça electrònica (URL) de la foto de perfil de l'usuari.

5.3.3.3 Protectora i Professional

Representen dos tipus d'usuari. El grup d'usuaris Protectores representen associacions protectores d'animals, mentre que el grup d'usuaris Professionals representen persones jurídiques relacionades amb l'àmbit animal.

- **web:** Adreça electrònica (URL) de l'associació o professional en qüestió.

5.3.3.4 Valuation

Representa el grup de valoracions entre usuaris.

- **value:** Valor de la puntuació.
- **comment:** Comentari addicional per justificar el valor de la puntuació.

5.3.3.5 Ad

Representa una classe base pels anuncis del sistema. Permet compartir els següents atributs entre les diferents implementacions.

- **tags:** Llista d'etiquetes relacionades amb l'anunci, útils pel motor de cerca.
- **photos:** Llista d'adreces electròniques de les imatges de l'anunci.

5.3.3.6 ServiceAd

Representa el grup d'anuncis sobre serveis.

- **title:** Títol del servei anunciat.
- **description:** Descripció del servei anunciat.
- **pricePerHour:** Preu l'hora del servei anunciat.

5.3.3.7 ProductAd

Representa el grup d'anuncis sobre productes.

- **title:** Títol del producte anunciat.
- **description:** Descripció del producte anunciat.
- **price:** Preu del producte anunciat.

5.3.3.8 AnimalAd

Representa el grup d'anuncis sobre mascotes.

- **name:** Nom de l'animal anunciat.
- **description:** Descripció de la mascota anunciada.
- **activityLevel:** Nivell d'activitat de la mascota anunciada.
- **birthDate:** Data de naixement de la mascota anunciada.
- **male:** Indica si la mascota anunciada és mascle o femella.
- **adoptionTax:** La taxa d'adopció de la mascota anunciada, en €.
- **weight:** Indica el pes de la mascota anunciada, en Kg.
- **personality:** Llista de trets de la personalitat de la mascota anunciada
- **mustKnow:** Informació addicional sobre la mascota anunciada.
- **deliveryInfo:** Llista d'informació de l'estat en el qual s'entrega la mascota anunciada.
- **breed:** Indica la raça de la mascota anunciada.
- **animalType:** Indica quin tipus d'animal és la mascota anunciada.

5.3.3.9 Dog

Representa el grup d'anuncis sobre gossos.

- **dogSize:** Indica la mida del gos anunciat.

5.3.3.10 Room

Representa el grup de conversacions entre usuaris.

5.3.3.11 Messages

Representa el grup de missatges entre usuaris.

- **text:** El contingut del missatge enviat.

6 Disseny

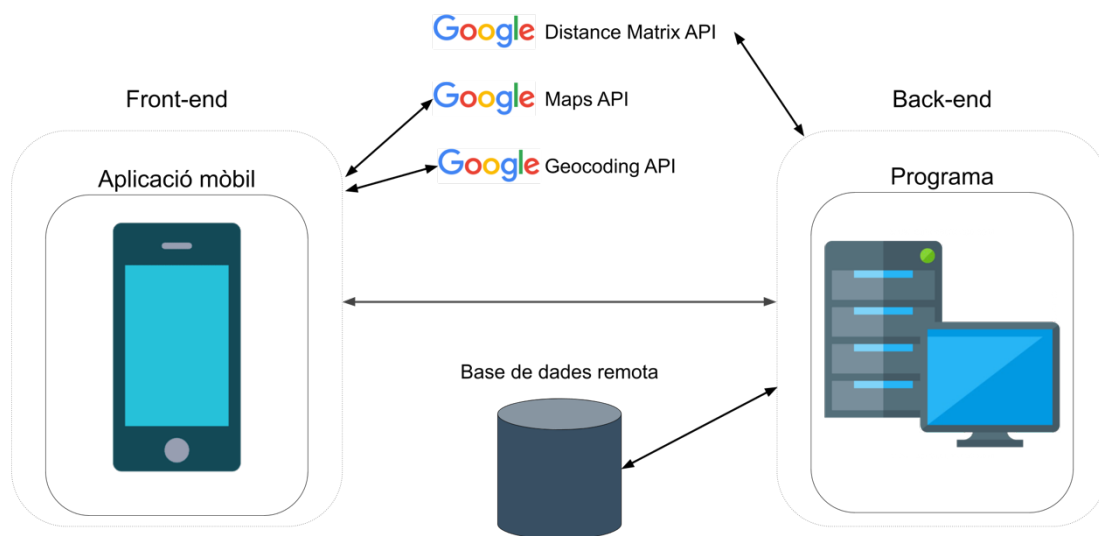
Un cop analitzats i especificats els requisits funcionals i no funcionals de l'aplicació, es procedeix amb la següent fase: el disseny del sistema software. L'objectiu d'aquesta fase és determinar els

components a desenvolupar suficients per a garantir el compliment dels requisits especificats anteriorment. A més, en aquest capítol es definirà l'arquitectura que el nostre sistema seguirà durant el desenvolupament.

6.1 Arquitectura

L'arquitectura d'aquest sistema es basa en dos components principals -tal com podem veure en la Il·lustració 6-: l'aplicació mòbil nativa per part del client (frontend), i el servidor (backend). Això vol dir que l'aplicació és la part amb la qual l'usuari interacciona; en canvi, el servidor és la part del sistema amb la que l'aplicació mòbil interacciona seguint diferents protocols, com ara HTTP¹². La idea central és que l'aplicació s'encarrega de saber que vol fer l'usuari en el sistema, i li transmet la informació al servidor, i aquest gestiona les dades i fa les modificacions necessàries.

A part dels components principals mencionats anteriorment, el sistema està compost per més elements. El primer element a destacar és la base de dades remota, la qual serà gestionada pel servidor principal del sistema. En segon i últim lloc, el sistema disposa d'accés a 3 APIs de Google¹³. El servidor necessita accedir al servei de *DistanceMatrix API* per calcular la distància i el temps entre dues coordenades. Simultàniament, el client necessita accedir als serveis *Maps API* i *Geocoding API* per mostrar per pantalla un mapa geogràfic dinàmic.



IL·LUSTRACIÓ 6: ABSTRACCIÓ DE L'ARQUITECTURA DEL SISTEMA

6.1.1 Front-end

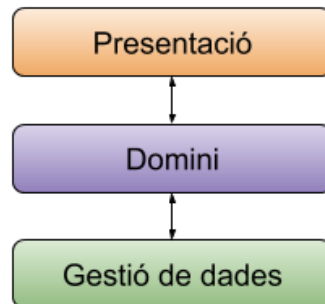
Quant a l'arquitectura lògica del front-end, s'aplicarà una basada en capes. Aquesta arquitectura aportarà al sistema una major seguretat, escalabilitat, sostenibilitat, i flexibilitat; ja que es divideix

¹² Protocol de transferència d'hipertext usat en la web

¹³ Companyia electrònica nord-americana de gran influència.

les responsabilitats de tot el sistema en diversos nivells. Addicionalment, aporta un desenvolupament més eficient, més re-usable, i és més fàcil d'afegir noves funcionalitats.

Per la part del front-end s'ha definit 3 capes principals – capa de presentació, capa de domini, i capa de gestió de dades- que seguidament s'explicaran les seves responsabilitats. En la següent il·lustració es mostra un diagrama de l'arquitectura esmentada.



IL·LUSTRACIÓ 7: ARQUITECTURA LÒGICA DE TRES CAPES

6.1.1.1 Capa de presentació

Aquesta capa té la principal responsabilitat de mostrar per pantalla el contingut de l'aplicació a l'usuari per a que aquest el pugui entendre. A més, s'encarregarà de traduir l'input¹⁴ de l'usuari a esdeveniments – s'explicarà en el següent apartat- perquè la capa de domini sàpiga com reaccionar.

Per actualitzar la interfície s'utilitzarà el patró Builder. Aquest patró aporta flexibilitat en el procés de la creació d'objectes en temps d'execució. Gràcies a això la interfície pot reaccionar segons l'estat que es rebí per part de la capa de domini, mostrant un contingut o un altre.

D'altra banda, per mostrar components per pantalla s'utilitzaran *widgets*. Els *widgets* són components específics del framework Flutter. Aquests components tenen l'objectiu de decidir com es veuran visualment depenent del seu estat o la seva configuració. S'explicarà més en detall aquest concepte en l'apartat 7.1.2.

6.1.1.2 Capa de domini

La capa de domini conté la lògica de negoci de l'aplicació. En altres paraules, reacciona mitjançant decisions lògiques a l'input rebut per part de la capa de presentació. A més mou i processa dades entre les dues capes contingents.

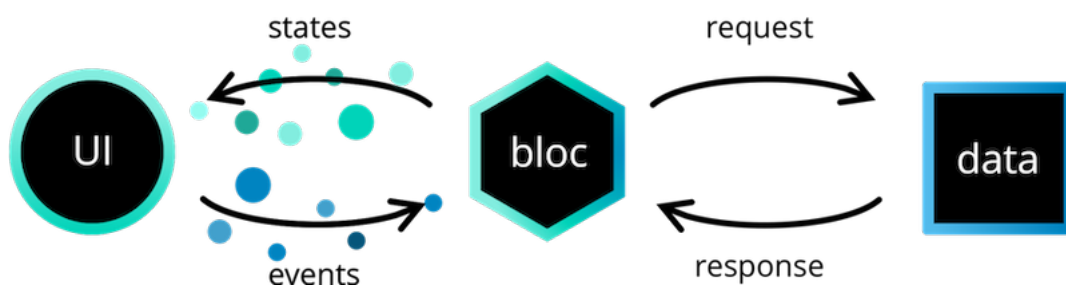
Així mateix, tot i que el servidor és el que s'encarrega del domini del sistema, l'aplicació mòbil ha de tenir prou lògica per a mantenir coherència entre les diferents vistes i mantenir l'estat de l'aplicació al reiniciar-la. Per aconseguir això, s'aplicarà el patró BLoC – Business Logic Component[25]-, que

¹⁴ Conjunt de dades que s'introdueixen en un sistema informàtic.

s'usa per gestionar l'estat d'una aplicació a base d'esdeveniments. A més, ens permet mantenir la separació entre la capa de presentació i la capa de domini.

Aquest patró va ser dissenyat amb 3 valors principals en ment:

- **Simple:** és fàcil d'entendre i pot ser utilitzat per diferents desenvolupadors de diferents nivells.
- **Potent:** la seva modularitat ajuda a crear sistemes complexos amb més facilitat.
- **Fàcil per testejar:** a part de ser fàcil de testejar per la seva modularitat, existeixen diverses llibreries que faciliten el *testing*¹⁵ d'aquests components.



IL·LUSTRACIÓ 8: ABSTRACCIÓ DEL PATRÓ BLOC, UTILITZAT EN EL FRONT-END

Com podem veure a la Il·lustració 8, usant aquest patró ens permet un flux bidireccional de la informació entre les 3 capes. Per entendre millor el patró i els seus principis s'explicarà cada component de la il·lustració:

- **UI:** La interfície de l'aplicació, és a dir la capa de presentació.
- **Esdeveniment:** Els esdeveniments són objectes que expressen la intenció de canviar l'estat del BLoC. Per exemple, hi haurà un esdeveniment per canviar l'idioma de l'aplicació, un altre per canviar l'aparença, etc. Aquests esdeveniments els rep el BLoC.
- **Estats:** Els estats també són objectes, però aquests representen la informació actual del BLoC.
- **BLoC:** El BLoC s'encarrega de canviar l'estat del component segons els esdeveniments rebuts. En cas necessari contactarà amb la capa de gestió de dades. Convé ressaltar que en una aplicació completa poden existir 50 o més BLoCs, per tant, el conjunt de tots els BLoCs representen la capa de domini.
- **Data:** Representa la capa de gestió de dades.

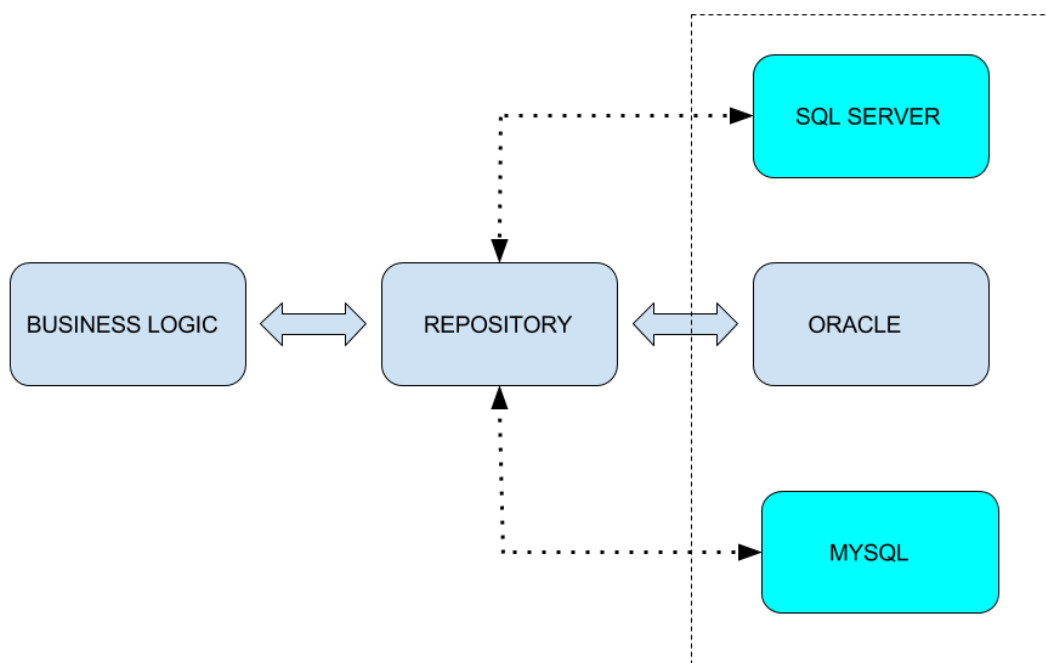
¹⁵ Fase dedicada a la realització de tests.

El patró BLoC ens ofereix diferents avantatges. En primer lloc, les actualitzacions previsibles dels estats faciliten la comprensió del flux de les dades dins de l'aplicació. En segon lloc, l'ús de funcions pures facilita testejar la lògica, i finalment la seva modularitat – no hi ha límit de nombre de BLoCs- permet usar estats de l'aplicació més precisos i detallats.

6.1.1.3 Capa de gestió de dades

L'objectiu principal de la capa de gestió de dades és, tal com diu el nom, gestionar les dades de l'aplicació. És a dir, s'encarrega d'enviar o demanar les dades corresponents al servidor o de gestionar la persistència d'aquestes en local. Per gestionar aquesta capa s'utilitzarà el patró Repositori.

El patró repositori proporciona una abstracció de dades, aconseguint així desacoblar totalment la capa de persistència a la lògica de negoci. D'aquesta manera – tot i que de moment no és el cas- ens permet tenir múltiples fonts de dades diferents, com es pot veure a la Il·lustració 9, sense afectar la usabilitat. Els repositoris solen tenir un aspecte semblant al d'una col·lecció: permeten agafar dades, afegir-ne d'una nova, modificar-la, o eliminar-la.



IL·LUSTRACIÓ 9: DIAGRAMA DEL PATRÓ REPOSITORI

6.1.2 Back-end

En l'anterior apartat s'ha dissenyat una arquitectura lògica de 3 capes pel front-end. En aquest apartat dissenyarem el back-end. Atès que el back-end no ha de disposar d'una capa de presentació en el nostre sistema, es dividirà l'arquitectura lògica d'aquest en dues capes: capa de domini i capa de gestió de dades.

6.1.2.1 Capa de domini

Aquesta capa, ja definida en l'apartat 6.1.1.2, disposarà de dos components principals: els controladors i els serveis.

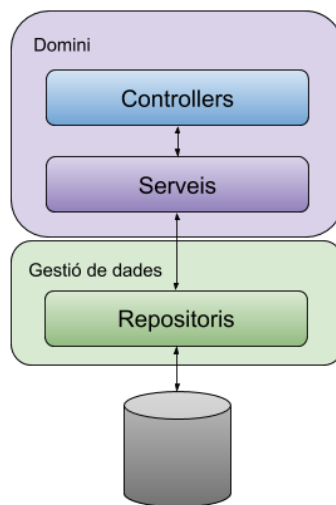
Primerament, els controladors tenen una responsabilitat ben definida: redirigir les peticions rebudes de l'API als serveis corresponents. A més, controlaran l'accés de les diferents funcionalitats i en cas de ser una funcionalitat restringida i l'usuari no està autoritzat, li denegarà l'accés. Aquest component no ha de disposar d'una lògica complexa, ja que serveix com a pont.

Per altra banda, els serveis disposen de la lògica de negoci del sistema. És a dir, modela el negoci de l'aplicació i contactarà amb la capa de gestió de dades quan sigui necessari fer operacions, actualitzar les dades existents, o guardar-ne de noves.

6.1.2.2 Capa de gestió de dades

De la mateixa manera que s'ha usat el patró Repositori en la capa de gestió de dades del front-end, s'usarà aquí també. D'aquesta manera si en un futur s'afegeixen més bases de dades no hi haurà problemes ni d'escalabilitat ni de dificultat.

Per acabar es mostra l'abstracció de l'arquitectura lògica del back-end en la següent il·lustració:



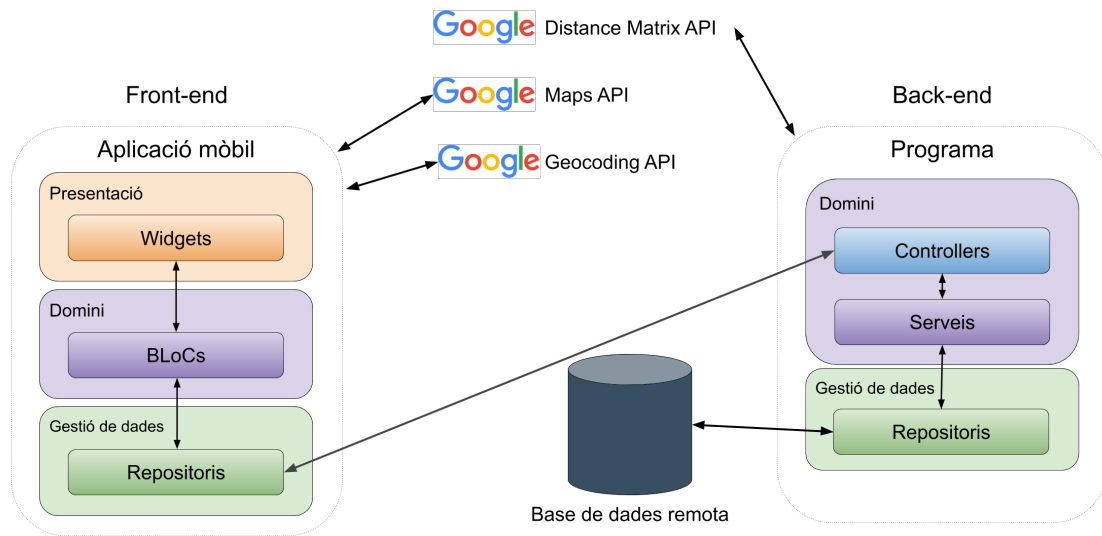
IL·LUSTRACIÓ 10: ABSTRACCIÓ DE L'ARQUITECTURA LÒGICA DEL BACK-END, BASADA EN CAPES

6.1.3 Sistema complet

Finalment, en la Il·lustració 11 es pot veure el disseny de l'arquitectura lògica i física del sistema complet. A més, durant la implementació del disseny se seguiran els principis SOLID¹⁶ [26], de manera que les capes només poden comunicar-se amb la capa més pròxima, mantenint així la

¹⁶ Representa 5 principis bàsics de la programació orientada a objectes

separació de cada una amb el seu propòsit. L'ús d'aquesta pràctica millora notablement la usabilitat, la sostenibilitat, i l'escalabilitat del sistema.



IL·LUSTRACIÓ 11: ARQUITECTURA LÒGIC I FÍSICA DEL SISTEMA COMPLET

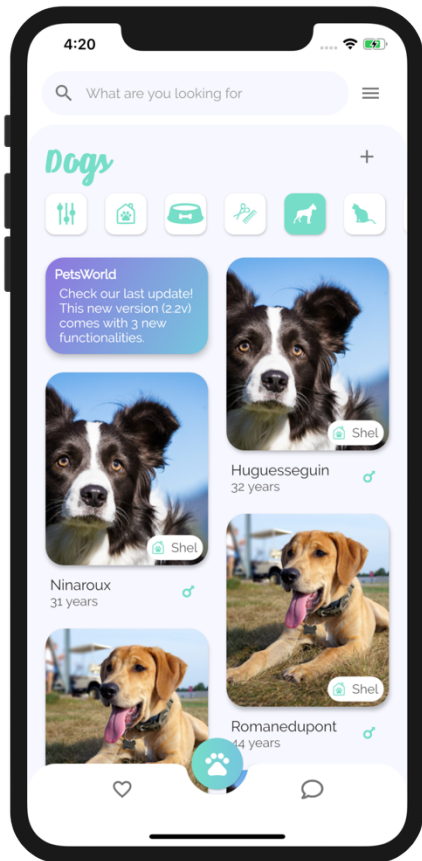
Més endavant es mostraran les classes que componen cada capa, amb els seus mètodes, els seus atributs i les associacions entre aquestes.

6.2 Disseny de l'aplicació (front-end)

6.2.1 Disseny de la UI

En aquest apartat es mostren diversos *mockups*¹⁷ que representen el disseny de les vistes amb les que interaccionaran els usuaris finals de l'aplicació mòbil, explicant per cada una d'elles les funcionalitats associades.

6.2.1.1 Pantalla principal



IL·LUSTRACIÓ 12: PANTALLA PRINCIPAL

La primera vista que veurà l'usuari a l'entrar a l'aplicació és la que es mostra a la Il·lustració 12. El disseny d'aquesta pantalla permetrà a l'usuari accedir a quasi totes les funcionalitats principals de l'aplicació.

Primerament, a la barra superior hi ha una barra de cerca, on l'usuari podrà buscar ràpidament les seves preferències, i a la seva dreta té un botó. Aquest botó és dinàmic, si l'usuari no ha iniciat sessió podrà accedir a les opcions de l'aplicació, en canvi, en el cas que l'usuari sí que hagi iniciat sessió, podrà accedir al seu perfil. Un usuari que hagi iniciat sessió i vulgui accedir a les opcions de l'aplicació ho podrà fer a través del seu perfil.

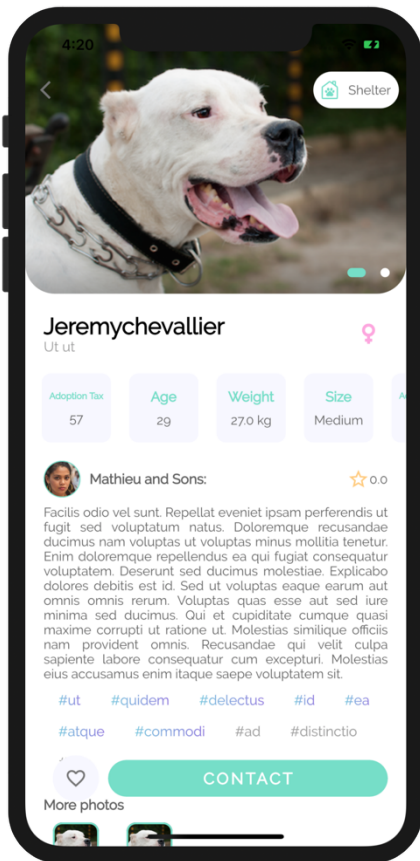
En segon lloc, en el cos de la vista es mostren els anuncis de mascotes, serveis, i productes; conjuntament amb una barra horitzontal per triar la categoria i un botó que permetrà pujar un anunci. A més, l'usuari podrà filtrar els anuncis de forma personalitzada segons els seus interessos. Per acabar, l'usuari serà capaç de veure les protectores més properes, junt amb

un mapa dinàmic que mostrarà la ciutat on se situa el centre, a quanta distància es troba en relació amb l'usuari, i a quants minuts en cotxe trigaria l'usuari a arribar.

Per acabar, a través de la barra inferior podrà accedir als anuncis preferits, tornar a la pantalla principal, o accedir als xats oberts.

¹⁷ Maqueta utilitzada per la demostració d'un disseny, d'una promoció o altres fins

6.2.1.2 Pantalla d'anunci



IL·LUSTRACIÓ 13: PANTALLA D'ANUNCI
podrà filtrar els anuncis, obtenint així només els anuncis que tinguin la mateixa etiqueta.

En el moment que l'usuari clica un anunci, l'aplicació obrirà la pantalla del anunci seleccionat. En aquesta vista, l'usuari tindrà totes les característiques de la mascota, servei, o producte a la seva disposició. En la Il·lustració 13 es pot veure un exemple d'un anunci d'una mascota.

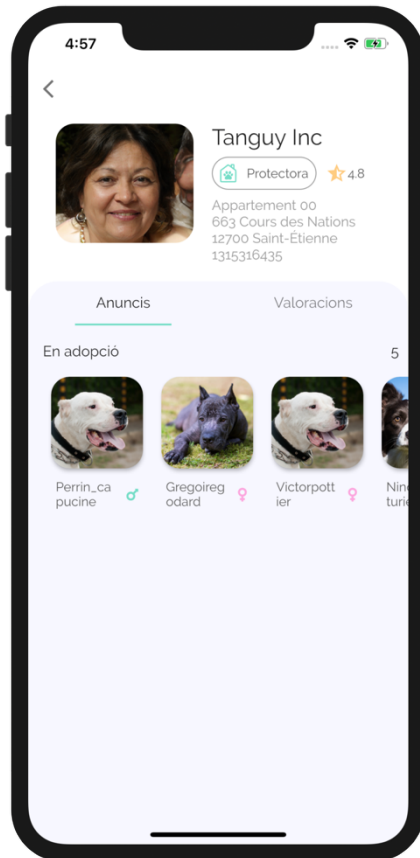
En primer lloc, es mostren les fotografies pujades amb l'anunci, juntament amb un component/chip que indica si l'anunci ha estat pujat per una protectora, un professional, o un particular.

Seguidament es mostren les característiques més importants de l'animal: el seu nom (*Jeremychevallier* en l'exemple), la seva raça (*Ut ut*), el seu sexe, la taxa d'adopció, l'edat, etc.

A continuació es mostra l'usuari que ha pujat l'anunci, junt amb la seva puntuació dins de l'aplicació, i una descripció sobre l'anunci en qüestió. A més a més, també es mostren les etiquetes. Si l'usuari clica en una etiqueta acolorida

Per acabar, a la barra inferior l'usuari podrà contactar amb l'usuari de l'anunci, o guardar aquest com a preferit.

6.2.1.3 Perfil d'usuari



IL·LUSTRACIÓ 14: PERFIL D'USUARI

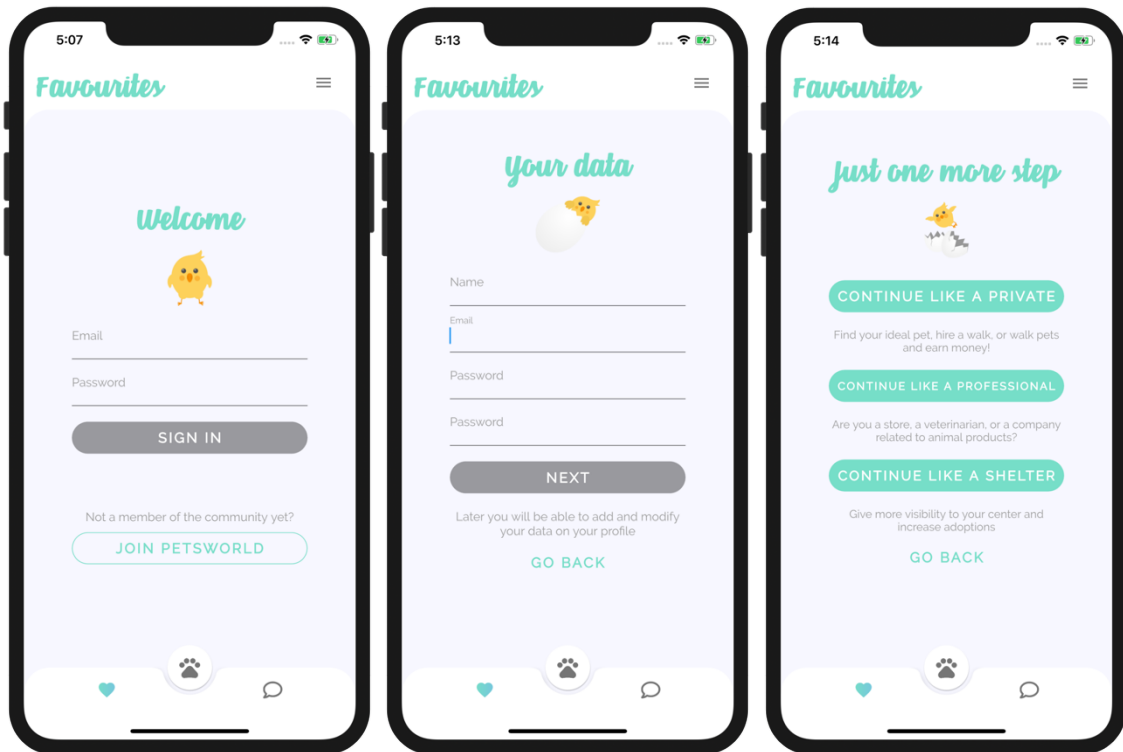
En el cas que l'usuari cliqui sobre el nom, o fotografia de l'usuari que ha penjat l'anunci, l'aplicació obrirà el perfil de l'usuari en qüestió. En el cas de la Il·lustració 14, es pot veure el cas d'una protectora.

En aquesta vista, l'usuari serà capaç de veure els detalls de l'usuari en qüestió: la imatge de perfil, nom, adreça, telèfon, tipus d'usuari (protectora, professional o particular) i la seva valoració mitja.

Sota la informació de l'usuari, hi ha dues seccions. La primera secció és on l'usuari podrà explorar quins anuncis ha penjat aquest usuari. En la segona secció podrà veure les valoracions fetes per els altres usuaris sobre aquest.

6.2.1.4 Inici de sessió i registre

L'aplicació no t'obligarà a iniciar sessió per ser utilitzada, però en el cas de que l'usuari vulgui accedir a alguna funcionalitat restringida serà necessari que l'usuari inici sessió. L'app està dissenyada perquè puguis iniciar sessió o registrar-te des de quasi totes les vistes.

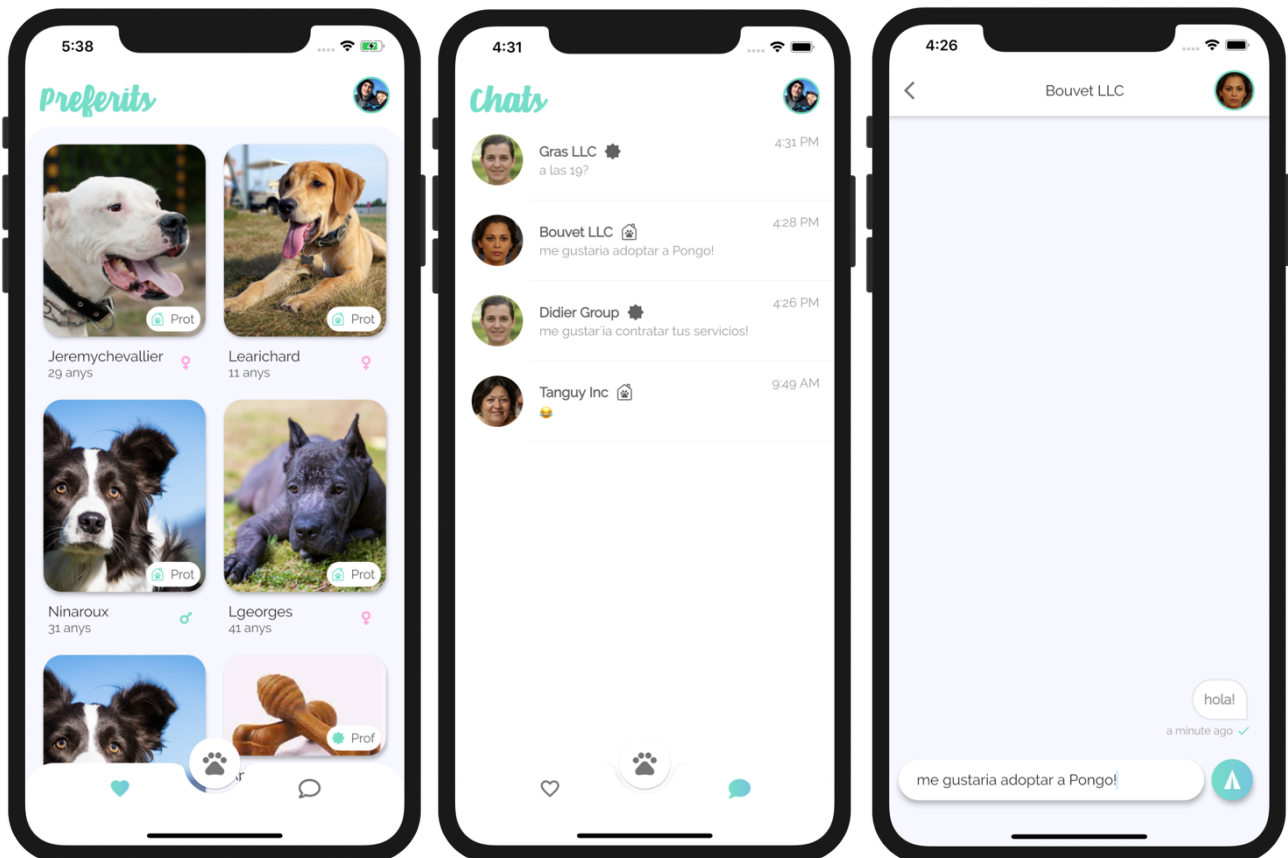


IL·LUSTRACIÓ 15: INICI DE SESSIÓ I REGISTRE

En la Il·lustració 15 es mostren tres mockups, d'esquerre a dreta: Inici de sessió, Primer pas del registre, i segon i últim pas del registre.

Per iniciar sessió serà necessari ingressar l'adreça electrònica i la contrasenya associada al conta. Tanmateix, per registrar-se s'ha decidit demanar el mínim de dades possibles per evitar el que s'anomena *pain points* (de l'anglès, punts de dolor). Els *pain points* és un concepte usat en disseny gràfic per referir-se als problemes amb els que l'usuari es troba i què la teva aplicació podria solucionar. En aquest cas concret, per millorar l'experiència d'usuari en un punt crític com és a l'hora de registrar un usuari, s'ha evitat el *pain point* d'haver d'omplir un formulari massa extens. D'aquesta manera, l'usuari només haurà d'omplir els camps obligatoris.

6.2.1.5 Pantalles de preferits i de xats

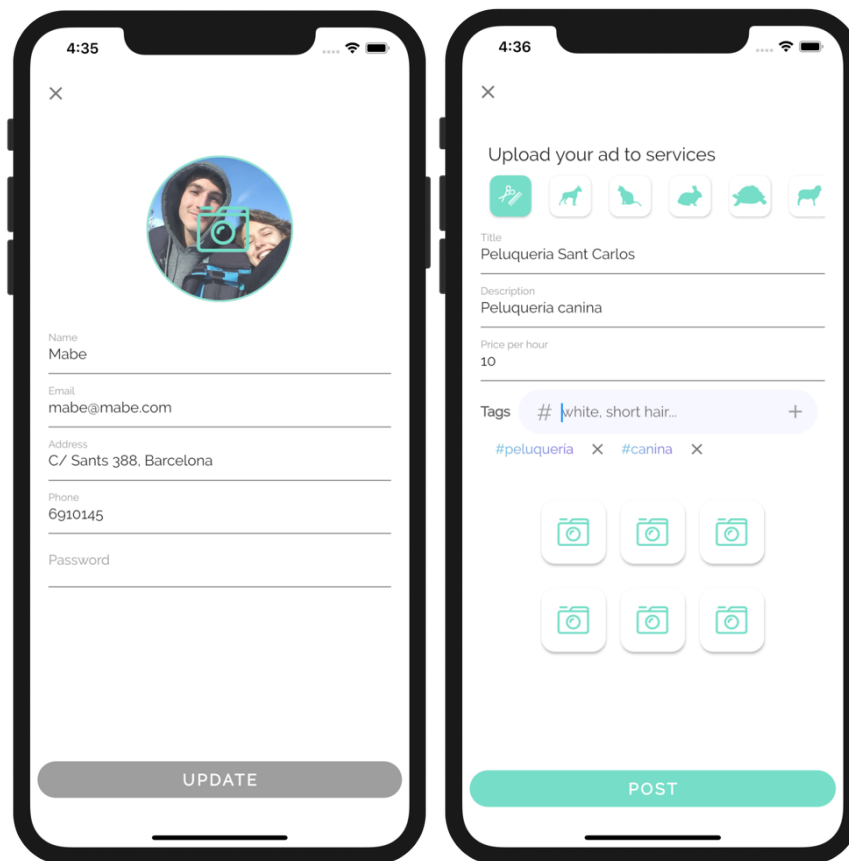


IL·LUSTRACIÓ 16: PANTALLES DE PREFERITS, DE XATS, I D'UN SOL XAT

Un cop iniciada la sessió, l'usuari serà capaç de visualitzar les diferents pantalles principals sense cap mena de restricció. Com es pot veure a la

Il·lustració 16 (d'esquerre a dreta: pantalla de preferits, pantalla de xats, pantalla d'un sol xat), l'usuari pot explorar els diferents anuncis guardats anteriorment, o entrar a un xat, també obert anteriorment. Per crear un xat, l'usuari haurà de navegar a l'anunci en concret i prémer el botó de *Contactar*. Un cop creat el xat, o un cop clicat a un xat ja existent, l'usuari serà redirigit a la pantalla d'un sol xat.

6.2.1.6 Pantalles de pujar anunci i editar perfil

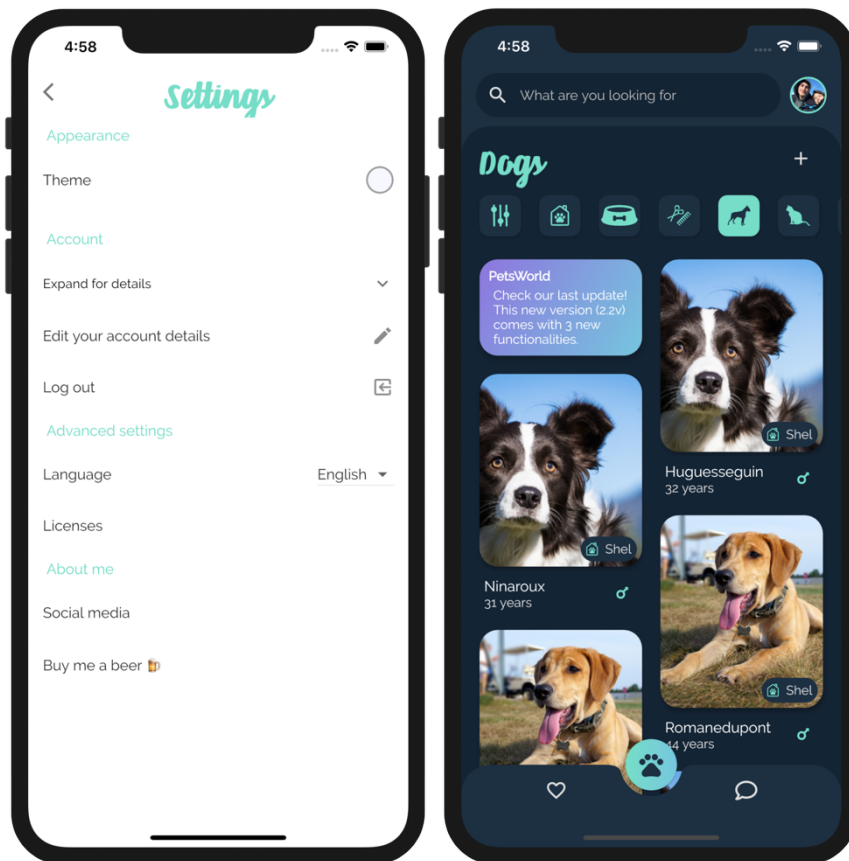


IL·LUSTRACIÓ 17: PANTALLS DE PUJAR ANUNCI I D'EDITAR PERFIL

Com es pot veure a la il·lustració anterior, el disseny de les pantalles amb formularis són senzills i directes; d'aquesta manera s'evita que l'usuari es perdi entre tants camps per a omplir. A l'esquerra de la il·lustració representa la pantalla on es pot editar les dades del perfil de l'usuari. Com s'ha comentat en l'apartat 6.2.1.4, a l'hora de registrar-se un usuari no té l'obligació a introduir totes les dades de la seva conta, per tant és important que l'usuari disposi d'una pantalla dedicada a l'edició d'aquestes. Per exemple: per fer-ne ús de la secció dedicada a visualitzar protectores properes a l'usuari, aquest haurà d'introduir primer una adreça vàlida.

A la dreta de la il·lustració anterior representa la pantalla per pujar un anunci. Les categories disponibles dependran del tipus d'usuari que sigui, per exemple un professional no serà capaç de pujar un anunci d'un animal. Tanmateix, els camps a omplir dependran totalment de la categoria seleccionada. Per exemple: si l'usuari vol crear un anunci d'un gos, aquest serà capaç d'indicar la mida de l'animal (entre gran, mitjà, i petit; que són les mides preestablertes); però aquest camp no existeix per a altres categories.

6.2.1.7 Pantalla d'opcions de l'aplicació



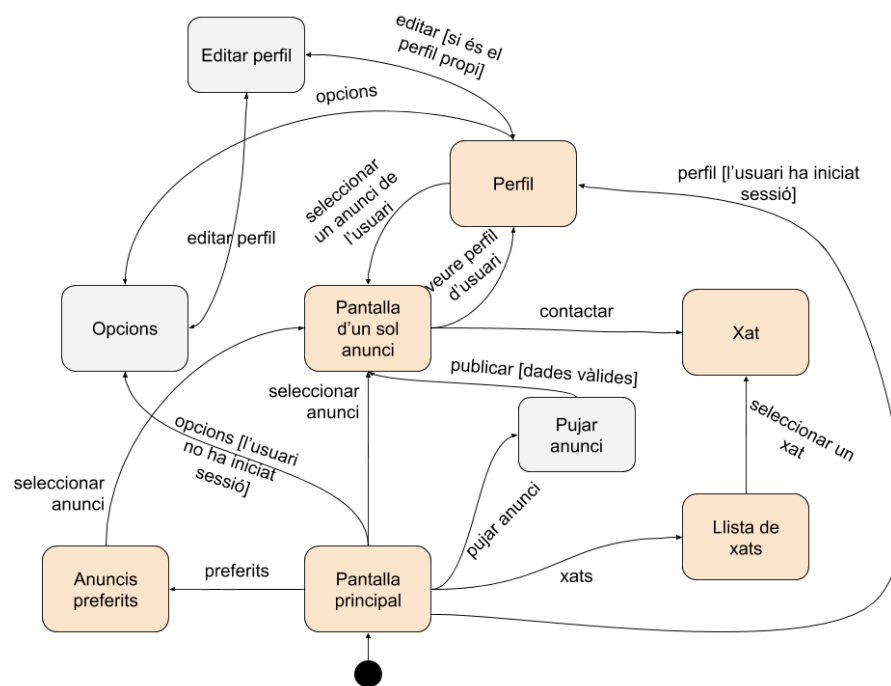
IL·LUSTRACIÓ 18: PANTALLA D'OPCIIONS DE L'APLICACIÓ I PANTALLA PRINCIPAL AMB TEMA FOSC

Per acabar, l'usuari disposarà d'una pantalla dedicada a les opcions de l'aplicació (Il·lustració 18, a l'esquerra). En aquestes opcions l'usuari podrà canviar l'idioma de l'app, veure les llicències que utilitza, visualitzar els detalls de la seva conta, sortir d'aquesta, entre d'altres.

Així mateix l'usuari podrà canviar l'aparença de l'aplicació, canviant el tema entre clar i fosc. Un exemple de la pantalla principal amb el tema fosc es pot visualitzar en la Il·lustració 18, a la dreta.

6.2.2 Mapa navegacional de la UI

A continuació a la Il·lustració 19 es mostren les diferents rutes de navegació entre les pantalles presentades anteriorment. Les pantalles principals estan representades en color i les pantalles secundàries estan representades en gris. A més, cada fletxa representa un click de l'usuari en el botó indicat pel comentari, i entre claudàtors les condicions que s'han de complir.

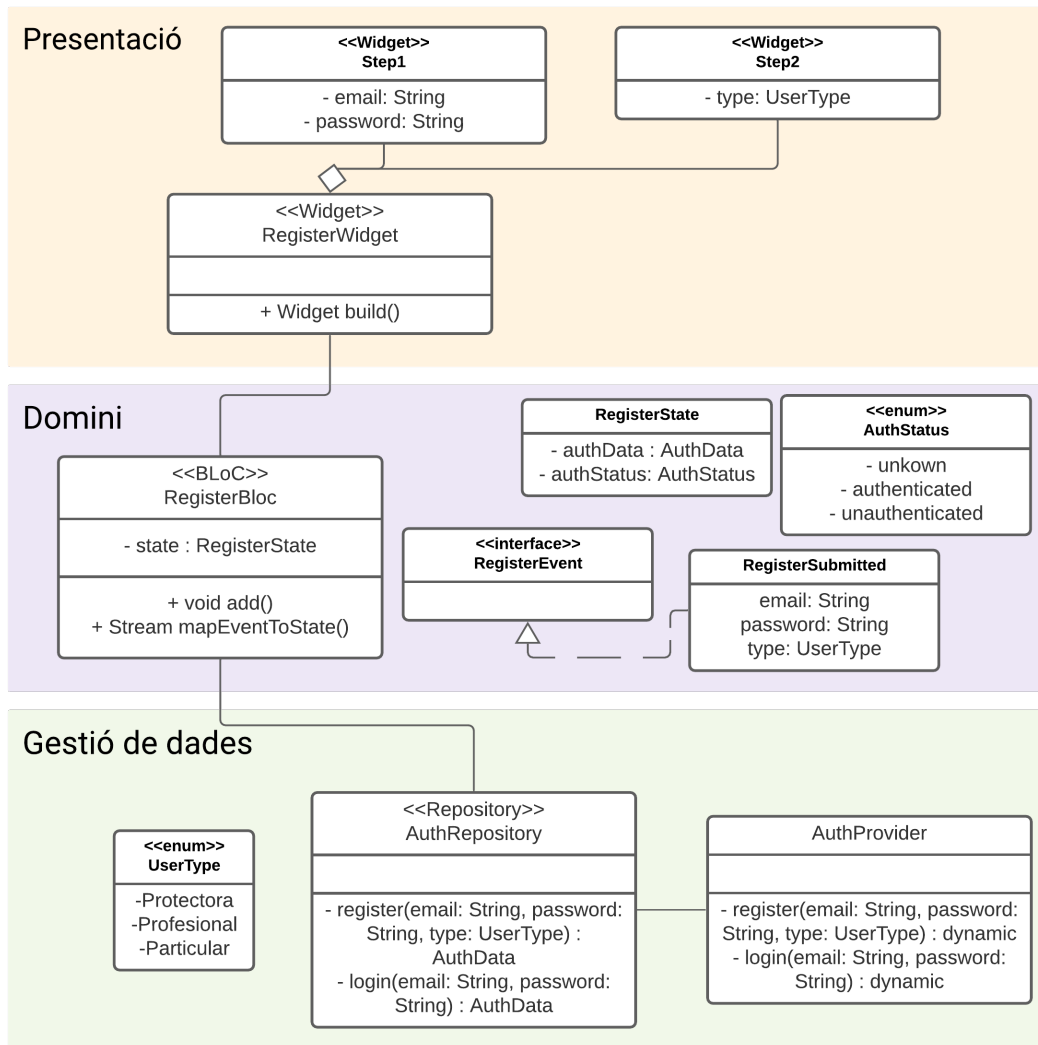


IL·LUSTRACIÓ 19: MAPA NAVEGACIONAL DE LES DIFERENTS PANTALLES DE L'APLICACIÓ

6.2.3 Disseny intern

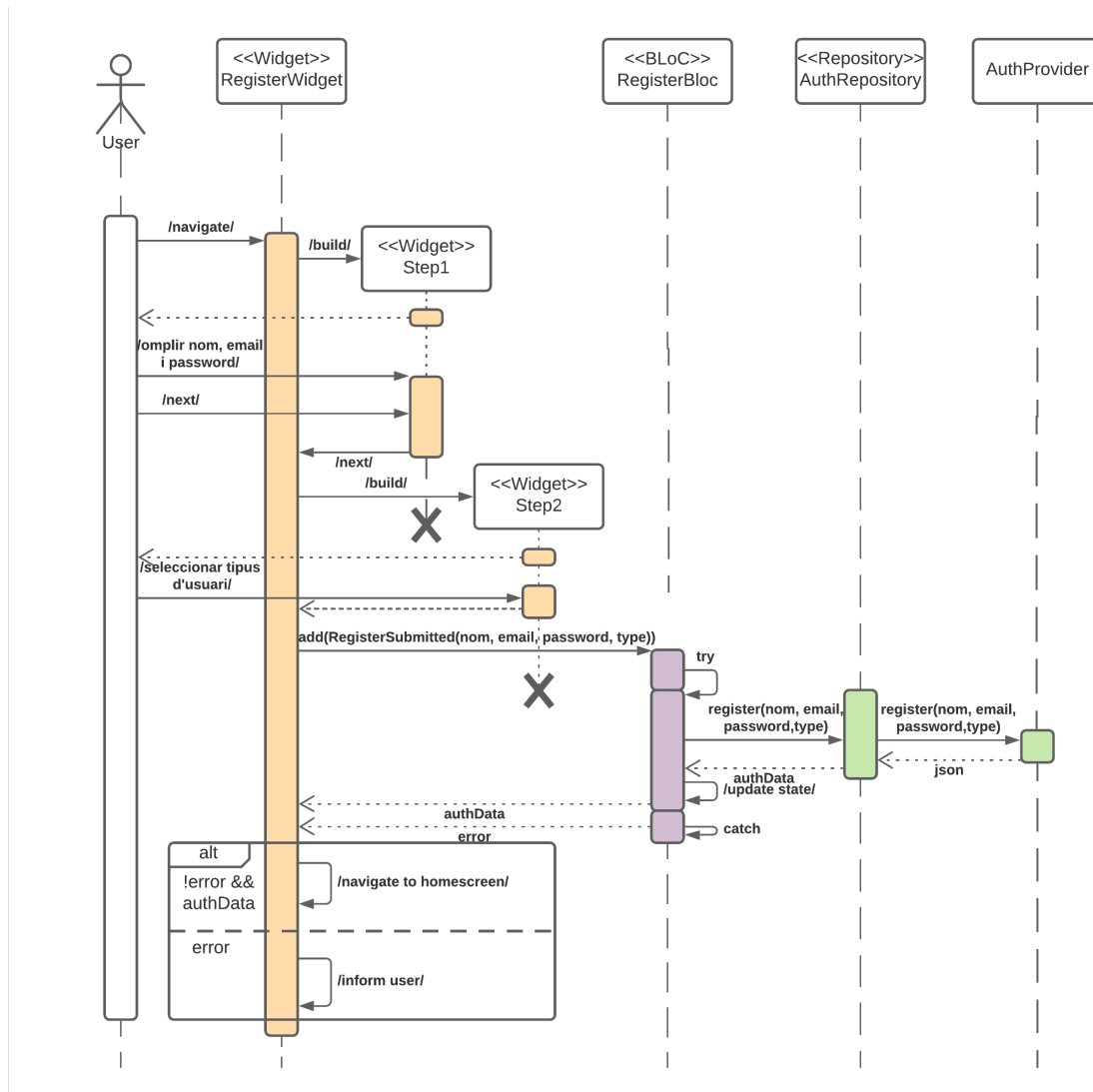
El desenvolupament d'un frontend no és gens trivial, sobretot quan es tracta d'una aplicació. En aquest apartat es mostra el disseny intern de les classes. Posat que fer un diagrama de totes les classes que intervenen en tot el frontend és innecessari i sobretot confús – ja que hi ha més de 50 classes per capa-, es mostrarà una simplificació amb una funcionalitat bàsica, com ara la de registrar-se. D'aquesta manera, amb l'exemple d'una funcionalitat bàsica, serà més comprensible l'arquitectura del sistema.

En la Il·lustració 20 s'exposen les diferents capes anteriorment dissenyades, amb les classes corresponents a cada una d'elles que han intervingut en aquesta funcionalitat.



IL·LUSTRACIÓ 20: DIAGRAMA DE CLASSES DEL REGISTRE EN EL FRONT END

En la següent il·lustració, la Il·lustració 21, es mostra el diagrama de seqüència per la funcionalitat de registrar-se. Aquest diagrama mostra gràficament les interaccions de l'usuari amb el sistema i el procés intern que fa l'aplicació com a resposta. Addicionalment, convé ressaltar que els colors de cada objecte representen la capa a la qual pertanyen.



IL·LUSTRACIÓ 21: DIAGRAMA DE SEQÜÈNCIA DE LA FUNCIONALITAT REGISTRAR-SE EN EL FRONTEND

Primerament, l'usuari navega al component de registrar-se en la interfície de l'aplicació. Un cop s'ha construït internament el component *RegisterWidget* aquest crearà un altre component -*Step 1*- on l'usuari proporcionarà la seva adreça electrònica, el nom i la contrasenya. Seguidament, quan l'usuari cliqui el botó *next*, *RegisterWidget* crearà el següent component -*Step 2*- on l'usuari indicarà el tipus de conta que vol crear. Un cop seleccionat, *RegisterWidget* afegirà un esdeveniment -*RegisterSubmitted*- al bloc *RegisterBloc*. Aquest bloc cridarà l'*AuthRepository* per registrar l'usuari. L'*AuthRepository* cridarà l'*AuthProvider* i aquest cridarà al servidor i retornarà la resposta corresponent. L'*AuthRepository* intentarà crear una resposta llegible pel sistema, i en cas de fallar farà saltar una excepció perquè el *RegisterBloc* l'agafi. A partir d'aquí hi ha dues variants: o bé no hi ha hagut cap error i l'usuari s'ha registrat correctament, o bé ha saltat un error. En el primer cas el component *RegisterWidget* activarà la navegació a la pantalla principal. En el segon cas, *RegisterWidget* mostrarà l'error a l'usuari.

6.3 Disseny del back-end

En aquest apartat es dissenyarà la part del servidor: el backend. En primer lloc es dissenyarà els mètodes públics que proporcionarà la seva API. En segon lloc s'exposarà el diagrama de classes i per finalitzar es mostrarà el diagrama de seqüència de la funcionalitat registrar-se, seguint l'exemple anterior, però en la part del backend.

6.3.1 Mètodes públics de l'API

Per aportar eficiència al sistema, s'ha escollit GraphQL com a idioma de consulta de l'API. Tal com indiquen a la seva web [13], GraphQL proporciona als clients el poder de demanar exactament les dades que necessiten, ni més ni menys. Així mateix, el client pot demanar en una sola petició múltiples recursos. A més, també asseguren que facilita l'evolució de l'API al llarg del temps, potenciant la sostenibilitat d'aquesta. En altres paraules, GraphQL és una alternativa a les API basades en REST¹⁸ amb l'objectiu de millorar la productivitat dels desenvolupadors i minimitzar la quantitat de dades que es transfereixen.

GraphQL té tres operacions bàsiques:

- **Query:** per realitzar consultes sobre les dades.
- **Mutation:** per modificar-les.
- **Subscription:** per subscriure's a un mètode, usant el protocol WebSocket. D'aquesta manera, el client rep notificacions en temps real sobre els canvis, com ara quan hi ha un nou missatge al xat.

Seguidament es mostraran les operacions públiques de cada una, però abans cal explicar el format. Primer s'esmenta el **nom** de l'operació, seguidament entre parèntesis s'esmenten els **paràmetres** que té la crida, i finalment, després dels dos punts s'esmenta l'**objecte** que torna l'operació. En l'apartat següent es mostrarà el diagrama de classes exposant què conté cada **objecte**. D'altra banda, quan un **paràmetre** o un **objecte** va seguit d'una exclamació, significa que aquest **paràmetre** o **objecte** és obligatori, no pot ser *null*. Altrament serà opcional.

6.3.1.1 Queries (consultes)

login(email: String!, password: String!): AuthData!

- Permet iniciar sessió en el sistema

¹⁸ Transferència d'estat representatiu, estil arquitectònic de software

getUser(id: String!): User

- Retorna l'usuari assignat al paràmetre *id*.

currentUser: User

- Retorna l'usuari assignat al *token*, prèviament havent iniciat sessió.

myRooms: [Room!]

- Retorna una llista amb els xats oberts de l'usuari.

getAd(id: String!): Ad

- Retorna l'anunci assignat al paràmetre *id*.

savedAds: [Ad!]

- Retorna els anuncis marcats com a preferits de l'usuari.

searchAds(filters: AdFilters!): [Ad!]

- Retorna els anuncis filtrats.

getCloseShelters(fromAddress: String): [DistanceData!]

- Retorna les protectores properes, indicant quant es triga en cotxe, i quants km de distància hi ha.

ads(category: Category!, first: Int!, after: String): AdConnection!

- Retorna els anuncis paginats de la categoria *category*.

6.3.1.2 Mutations

createUser(userInput: UserInput!): AuthData!

- Permet crear un nou usuari.

updateUser(userInput: UserInputOptional!): User!

- Permet actualitzar les dades del teu usuari.

createAnimalAd(adInput: AnimalAdInput!): AnimalAd!

- Permet crear un anunci d'animals.

deleteAnimalAd(id: String!): AnimalAd!

- Permet eliminar un anunci animal.

createProductAd(adInput: ProductAdInput!): ProductAd!

- Permet crear un anunci de productes.

deleteProductAd(id: String!): ProductAd!

- Permet eliminar un anunci de productes

createServiceAd(adInput: ServiceAdInput!): ServiceAd!

- Permet crear un anunci de serveis.

deleteServiceAd(id: String!): ServiceAd!

- Permet eliminar un anunci de serveis

valueateUser(input: ValuationInput): User!

- Permet avaluar un perfil extern.

removeValuation(id: String!): User!

- Permet eliminar una avaluació realitzada prèviament.

saveAd(id: String!): [Ad!]

- Permet guardar un anunci com a preferit.

unsaveAd(id: String!): [Ad!]

- Permet eliminar un anunci com a preferit.

createMessage(toUser: String!, text: String!, ad: String): Room!

- Permet enviar un missatge a un usuari.

6.3.1.3 Subscriptions

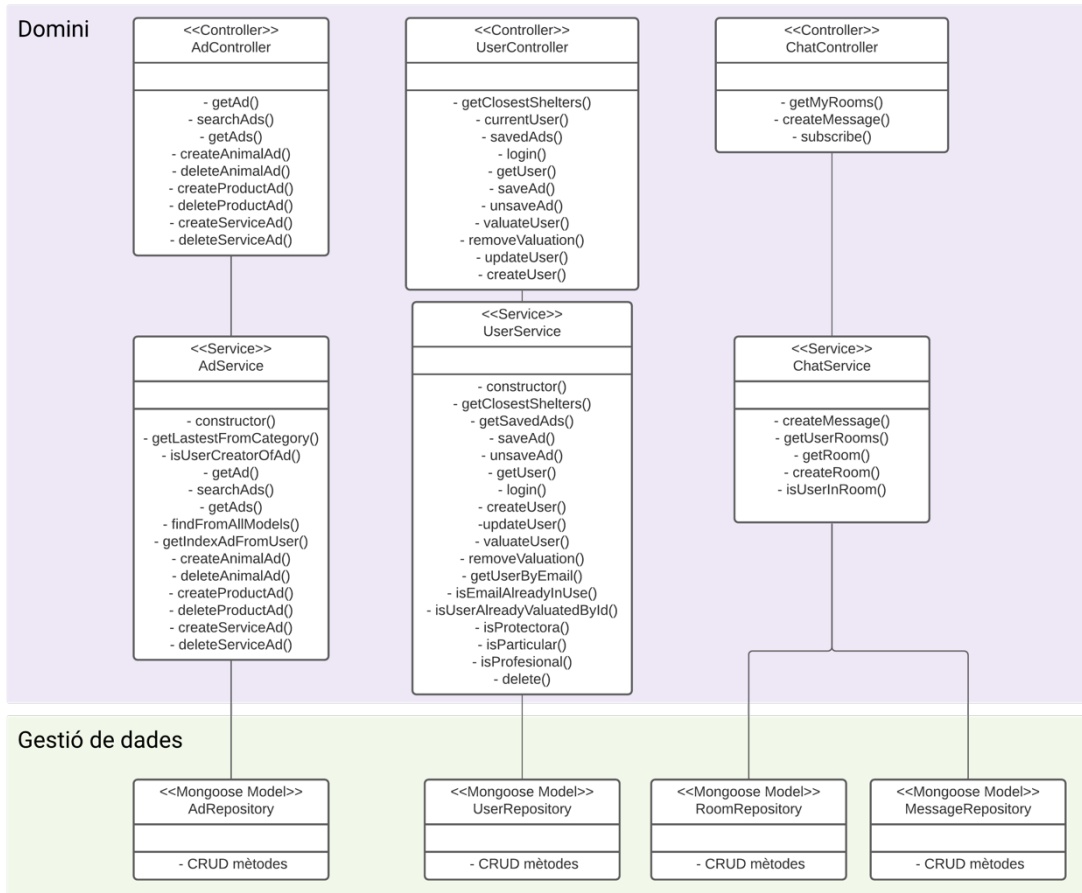
messageSent(roomId: String!): Message

- Permet subscriure's a un xat, rebent així notificacions quan hi ha un nou missatge.

6.3.2 Diagrama de classes del back-end

Gràcies a l'abstracció que aporta l'ús de llibreries com Mongoose [27] – a l'hora de gestionar els models i la base de dades- o ApolloServer [28] – a l'hora de gestionar GraphQL i la API- el diagrama

de classes del back-end és molt simple. Bàsicament es pot resumir amb els controladors, els serveis i els repositoris, com es pot veure a la Il·lustració 22.

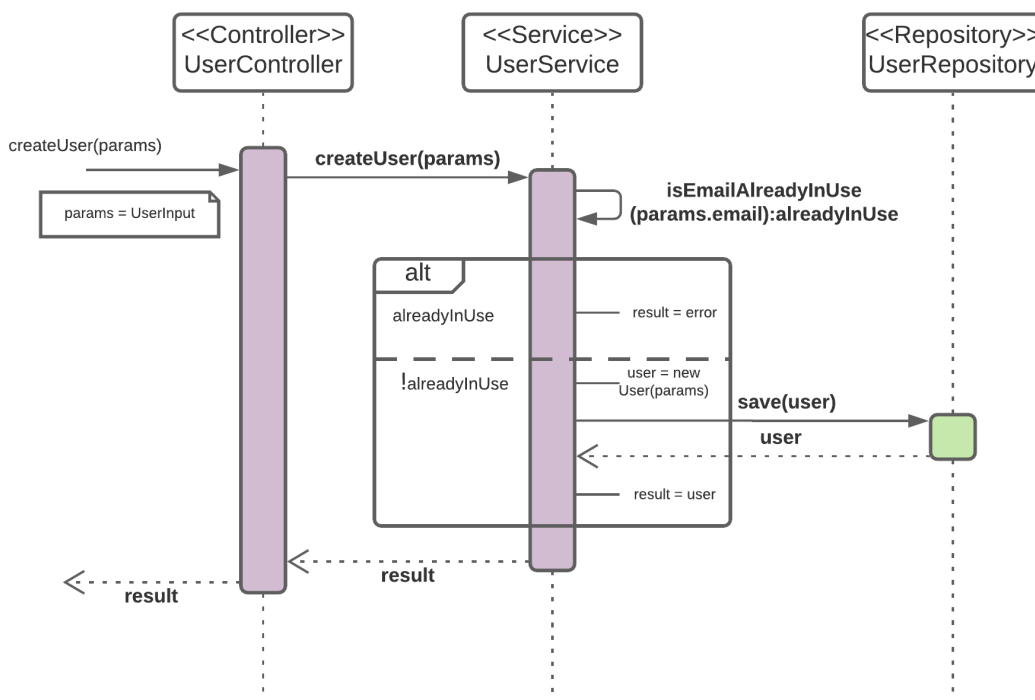


IL·LUSTRACIÓ 22: DIAGRAMA DE CLASSES DEL BACK-END

6.3.3 Diagrama de seqüència

Seguint amb l'exemple de la funcionalitat de registrar-se, començat anteriorment en el frontend, es finalitzarà mostrant el disseny del servidor. En el diagrama de seqüència del frontend la classe *AuthProvider* envia una petició al servidor per intentar registrar l'usuari.

En la part del servidor GraphQL redirigirà la petició al controlador corresponent depenent el mètode rebut –6.3.1- a través de la petició. Seguint amb l'exemple de registre, la petició la rebrà el controlador d'usuaris. Aquest delegarà la lògica al *UserService*, ja que no és responsabilitat del controlador. El servei revisarà que el correu no està ja registrat, i si no és el cas cridarà al *UserRepository* perquè guardi el nou model. La lògica del repositori és implícita a través d'una llibreria que es comentarà en el següent apartat.



IL·LUSTRACIÓ 23: DIAGRAMA DE SEQÜÈNCIA DEL BACK-END, RELATIU A LA FUNCIONALITAT DE REGISTRAR-SE

6.4 Esquema de la base de dades

Per finalitzar aquest capítol es mostra seguidament l'esquema de la base de dades. Aquest esquema es defineix a través de la llibreria comentada anteriorment, Mongoose, ja que és la responsable de generar les classes repositori i contactar amb la BD.

Particular (ID, name, thumbnail, email, address, phone, password, valuations)

Check {email} is UNIQUE

Check {name} is NOT NULL

Check {email} is NOT NULL

valuations referencia Valuations

Protectora (ID, name, thumbnail, email, address, phone, password, valuations, web)

Check {email} is UNIQUE

Check {name} is NOT NULL

Check {email} is NOT NULL

valuations referencia Valuations

Professional (ID, name, thumbnail, email, address, phone, password, valuations, web)

Check {email} is UNIQUE

Check {name} is NOT NULL

Check {email} is NOT NULL

valuations referencia Valuations

Valuation (ID, value, comment, author)

Check {value} is NOT NULL

Check {comment} is NOT NULL

Check {author} is NOT NULL

autor referencia Professional o Particular o Protectora

SavedAd (ID, ad, user)

Check {ad} is NOT NULL

Check {user} is NOT NULL

user referencia Professional o Particular o Protectora

ad referencia ServiceAd o ProductAd o AnimalAd

Message (ID, text, sender)

Check { text } is NOT NULL

Check { sender } is NOT NULL

sender referencia Professional o Particular o Protectora

Room (ID, user1, user2, messages)

Check { user1 } is NOT NULL

Check { user2 } is NOT NULL

user1 referencia Professional o Particular o Protectora

user2 referencia Professional o Particular o Protectora

ServiceAd (ID, tags, photos, title, description, priceHour, creator)

Check { tags } is NOT NULL

Check { photos } is NOT NULL

Check { title } is NOT NULL

Check { description } is NOT NULL

Check { priceHour } is NOT NULL

Check { creator } is NOT NULL

creator referencia Professional o Particular

ProductAd (ID, tags, photos, title, description, price, creator)

Check { tags } is NOT NULL

Check { photos } is NOT NULL

Check { title } is NOT NULL

Check { description } is NOT NULL

Check { price } is NOT NULL

Check { creator } is NOT NULL

creator referencia Professional

AnimalAd (ID, tags, photos, name, description, adoptionTax, activityLevel, birthDate, male, weight, personality, mustKnow, deliveryInfo, breed, type, creator)

Check { tags } is NOT NULL

Check { photos } is NOT NULL

Check { name } is NOT NULL

Check { description } is NOT NULL

Check { adoptionTax } is NOT NULL

Check { activityLevel } is NOT NULL

Check { birthDate } is NOT NULL

Check { male } is NOT NULL

Check { weight } is NOT NULL

Check { personality } is NOT NULL

Check { deliveryInfo } is NOT NULL

Check { type } is NOT NULL

Check { creator } is NOT NULL

creator referencia Protectora o Particular

7 Implementació

Un cop finalitzada la fase de disseny ja es pot començar amb la fase d'implementació. En aquest apartat es detallen els aspectes tècnics del procés: enfocaments, eines o llibreries, patrons utilitzats, i metodologia.

Seguint la metodologia planificada anteriorment – apartat Metodologia i gestió del projecte-, la implementació es farà a partir d'*Sprints*, dividint els casos d'ús – definits en l'apartat 5.1.2- entre ells i realitzant de forma periòdica reunions retrospectives.

7.1 Implementació del front-end

7.1.1 Versions d'eines

A partir del 2 de Novembre del 2020 Android aplicarà un nou requeriment per a pujar l'aplicació al seu mercat – Google Play- [29]. Android demana que totes les noves aplicacions treballin amb Android 10, és a dir amb l'Android SDK 29. Ja que Android permet definir un rang de versions de compatibilitat s'ha escollit com a versió màxima la 29, i la mínima la 16. Amb aquest rang de versions de compatibilitat **l'aplicació serà compatible amb el 99,8% de dispositius Android**; segons la següent taula extreta de dades oficials de Google.

Versió de la plataforma Android (Nivell d'API)	Distribució
Android 4.0 "Ice Cream Sandwich" (15)	0.2 %
Android 4.1 - 4.3 "Jelly Bean" (16, 17, 18)	1,7 %
Android 4.4 "KitKat" (19)	4 %
Android 5.0 – 5.1 "Lollipop" (21, 22)	9,2 %
Android 6.0 "Marshmallow" (23)	11,2 %
Android 7.0 – 7.1 "Nougat" (24, 25)	12,9 %
Android 8.0 "Oreo" (26)	7,3 %
Android 8.1 "Oreo" (27)	14 %
Android 9 "Pie" (28)	31,3 %
Android 10 (29)	8,2 %

TAULA 31: DISTRIBUCIÓ DE LA VERSIÓ D'ANDROID EN DISPOSITIUS, A DATA DE 10 D'ABRIL DE 2020

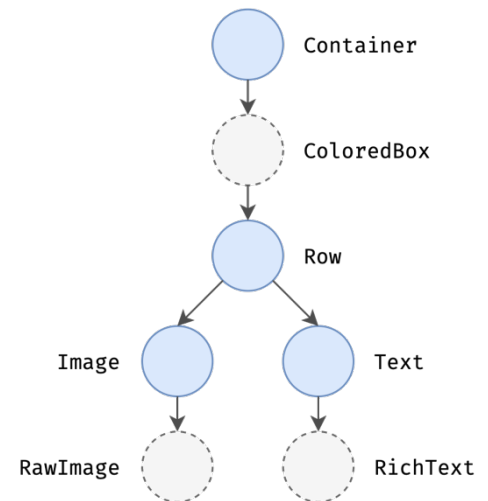
Quant a Flutter, durant l'estiu del 2020 l'equip de Google ha publicat una nova versió del framework, la 1.20, incloent-hi certes novetats [30]. Entre aquestes es pot destacar millores en el rendiment, o la reducció de la mida de l'aplicació via *tree shaking*¹⁹. Per aquest fet, s'usarà la versió de Flutter 1.20 durant la implementació de l'aplicació.

7.1.2 Filosofia de Widgets

Flutter defineix *widgets* com a unitats de composició. En altres paraules, els *widgets* són elements bàsics de la interfície d'usuari d'una aplicació, sent cada *widget* una declaració immutable d'una part de la interfície.

Un conjunt de *widgets* formen una composició jeràrquica semblant a la d'un arbre, com es pot veure a la Il·lustració 24. Típicament l'arrel d'aquest graf serà un contenidor que proporciona Flutter, anomenat *MaterialApp* o *CupertinoApp*, depenent a quin disseny apunta l'aplicació en concret. Addicionalment, un *widget* pot tenir estat propi (que alguna propietat del *widget* canvi respecte al temps) o pot no tenir-ne, com ara una icona o una etiqueta.

Widgets



IL·LUSTRACIÓ 24: COMPOSICIÓ JERÀRQUICA DE WIDGETS, FONT: FLUTTER.DEV

7.1.3 Tecnologies usades i aplicades

Durant la implementació s'ha fet ús de diferents tecnologies i eines. En aquest apartat s'explica quines s'han fet servir i perquè l'ús d'aquestes millora el valor de l'aplicació.

7.1.3.1 Immutabilitat i serialització

A l'hora de programar és bona pràctica treballar amb objectes immutables, ja que ofereixen certs avantatges respecte als objectes mutables. Convé ressaltar que un objecte és immutable quan el seu estat no canvia després de la seva inicialització.

Atès a la immutabilitat dels objectes, el desenvolupador no s'ha de preocupar per si algun mètode canviarà de valor l'objecte sobre el qual està treballant. Això comporta en un codi més predictiu i segur; reduint així la quantitat de possibles *bugs*²⁰.

¹⁹ En informàtica, *tree shaking* és una tècnica d'eliminació de recursos inactius que s'aplica a certs llenguatges

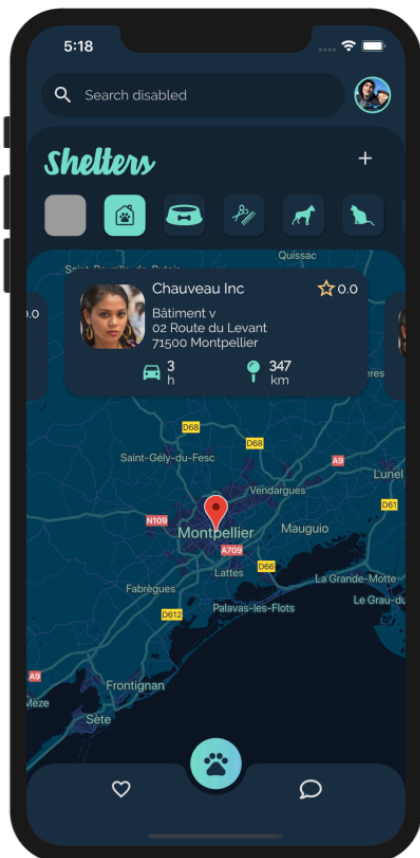
²⁰ Un bug de software és un error, manca o falla en el programa o sistema que causa un comportament inesperat

En conseqüència, s'ha trobat adient l'ús d'una llibreria que ens proporcioni aquesta habilitat, com ara *built_value* [31]. Aquesta llibreria no només ofereix tipus immutables, sinó que també proporciona suport per la serialització²¹ de les dades.

Adicionalment, aquesta llibreria utilitza el patró de disseny Builder per la "modificació" dels objectes. Atès que aquest patró permet construir un objecte pas a pas, l'ús d'aquest permet reconstruir de forma eficient un objecte només modificant una o més propietats. Resultant així en un bon acompanyant de la immutabilitat.

7.1.3.2 Mapa dinàmic

Anteriorment s'ha especificat la funcionalitat del sistema de mostrar les protectores properes, indicant el temps estimat d'arribada en un automòbil i la distància a recórrer. Com a fi de fer més aparent i útil la interfície gràfica d'aquesta funcionalitat s'ha cregut adient mostrar un mapa dinàmic que mostri la zona geogràfica en la qual es troba la protectora en concret.



Per implementar aquesta idea s'ha fet ús de la llibreria *google_maps_flutter* [32] desenvolupada per l'equip de Flutter. Aquesta llibreria proporciona un widget amb varies opcions per indicar com es comportarà el mapa, o bé per limitar certes funcionalitats del widget. Aquesta llibreria contacta de forma implícita amb l'API de Google Maps SDK per Android i Google Maps SDK per iOS.

Adicionalment, s'ha fet ús de la llibreria *geocoder* [33] que permet cridar a l'API de Google Geocoding per transformar una adreça a coordenades i viceversa. Primerament es va intentar obtenir aquesta funcionalitat a partir dels serveis de geocodificació nadius del dispositiu, però malauradament el servei de geocodificació dels dispositius iOS és menys flexible que el que ofereix Google. Per tant, l'ús d'aquesta eina permet als usuaris més flexibilitat a l'hora d'escriure la seva adreça.

IL·LUSTRACIÓ 25: DEMOSTRACIÓ GRÀFICA DEL MAPA DINÀMIC DE L'APLICACIÓ

Finalment, en la Il·lustració 25 es pot veure el resultat gràfic de la implementació d'aquest mapa dinàmic.

²¹ La serialització és el procés de transformar un objecte a una llista de bytes, com a objectiu de guardarlo.

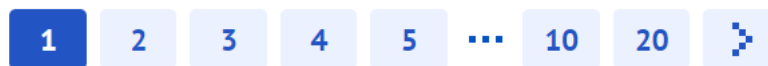
Simultàniament quan l'usuari navegui pels diferents perfils de protectores el mapa s'actualitzarà amb la nova direcció mitjançant una animació.

7.1.3.3 Minimització de recursos i desplaçament infinit

No tots els usuaris disposen d'una connexió a internet potent o ràpida, per tant intentar minimitzar la quantitat de bytes que necessita transmetre o rebre l'aplicació per a ser usable és un factor important.

Tenint en compte aquest fet, es paginaran els anuncis de mascotes, productes i serveis com a objectiu de minimitzar els recursos. Val la pena explicar que la paginació és el procés de dividir un document en diferents pàgines. D'aquesta manera quan l'usuari navegui per les diferents categories l'aplicació no haurà de demanar al servidor tots els anuncis de la categoria en concret, sinó només els últims 10 per exemple. Finalment, en un apartat posterior s'explicarà quin algorisme s'ha implementat per part del servidor per permetre l'ús d'anuncis paginats en la part del client.

La forma més senzilla d'implementar aquest comportament per part del front-end seria mostrant un índex de pàgines com es mostra a la Il·lustració 26.



IL·LUSTRACIÓ 26: EXEMPLE D'UN ÍNDEX DE PÀGINES [34]

Tot i que aquest disseny és útil, obliga a l'usuari a clicar sobre la següent pàgina per carregar els següents anuncis. A més no segueix el flux natural del desplaçament vertical a l'hora de navegar. Per aquests motius, s'ha trobat adient implementar un desplaçament infinit, seguint així amb el flux de l'usuari i eliminant així clics innecessaris.

Adicionalment, Flutter proporciona un *widget* – *ListView.builder()*– que permet la creació d'una llista infinita a través del patró Builder, comentat anteriorment, de manera senzilla.

7.1.3.4 Marcador de contingut



IL·LUSTRACIÓ 27: DEMOSTRACIÓ DELS MARCADORS DE CONTINGUT EN L'APLICACIÓ

En termes d'experiència d'usuari és important mostrar a l'usuari ràpidament el contingut de l'aplicació. Això no sempre és possible, ja sigui per falta d'infraestructura, falta de recursos, o falta d'optimització. Tenint això en compte és bona idea mostrar a l'usuari, d'alguna manera intuïtiva, que el contingut de l'aplicació s'està carregant. No només això, seria ideal també poder mostrar a l'usuari com es veurà distribuït el contingut que s'està carregant.

La solució a aquests problemes és l'ús del component *ContentPlaceholder* o marcadors de contingut. Aquest component es tracta de mostrar a l'usuari un contenidor visible, de mida semblant al contingut a mostrar. Per acabar d'entendre el concepte, a la Il·lustració 27 s'exposen els marcadors de contingut mentre es carreguen els anuncis sol·licitats. La idea rere això és que els usuaris són més pacients i creuran que el sistema és més ràpid i fluid si saben què està passant i poden anticipar el contingut abans que aquest es carregui.

7.1.3.5 Aparença de l'aplicació

Seguint en la línia d'experiència d'usuari s'ha optat per oferir dos temes d'aparença en l'aplicació, un clar i un fosc, sent el clar el tema inicial. Aquesta funcionalitat s'ha implementat amb l'objectiu de reduir la fatiga ocular, ja que un tema fosc redueix la llum blava de la pantalla, produint així un efecte més plaent. A més, oferir l'opció a l'usuari a triar la seva preferència personal és un bon afegit per satisfer-lo. En les següents il·lustracions es poden veure els dos temes en la pantalla d'un mateix anunci.



IL·LUSTRACIÓ 28: POSSIBLES APARENCES DE L'APLICACIÓ

7.2 Implementació del back-end

7.2.1 Apollo Platform i Express

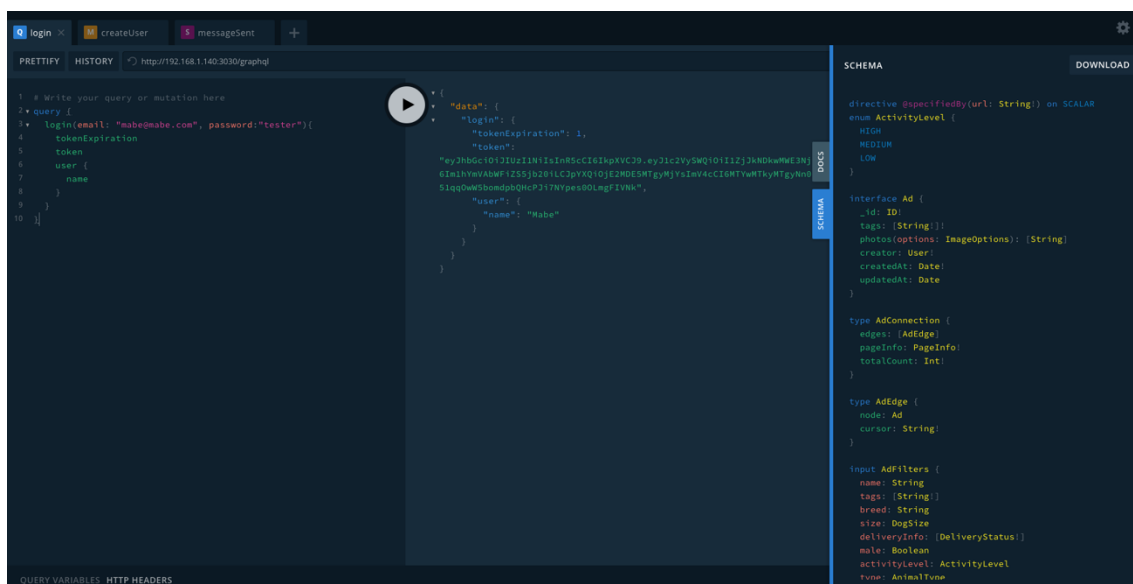
El backend d'aquest projecte ha estat desenvolupat fent servir el framework Express. Express és una infraestructura web ràpida, minimalista i flexible per a NodeJS. L'ús d'aquesta eina ha aportat certs avantatges durant la implementació del backend: fàcil d'integrar serveis de tercers, fàcil d'aprendre, és sostenible, entre d'altres. A més convé ressaltar que té una gran comunitat darrere donant suport.

D'altra banda, Express és més enfocat a la creació d'APIs REST. És per això que s'ha integrat junt amb Apollo Platform. Apollo Platform és una implementació de GraphQL fàcilment integrable a Express. Apollo Platform aporta moltes funcionalitats integrades que permeten treure una gran complexitat a la configuració del servidor. A més, ajuda a la gestió d'errors, cache, subscripcions, etc.

Per les raons comentades, el servidor del backend s'ha desenvolupat fent servir la versió 2.16.1 de la llibreria *apollo-server-express*, que fa ús d'Apollo Platform integrat a Express.

7.2.2 GraphQL Playground

GraphQL Playground [35] és un IDE de GraphQL que millora el flux de treball durant el desenvolupament. L'ús d'aquesta eina exposa una interfície del servidor on pots cridar als mètodes públics del teu servidor de manera ràpida i senzilla. Gràcies a aquest IDE no fa falta l'ús d'eines com Postman o curl per provar les funcionalitats del servidor. En la Il·lustració 29 es pot veure la interfície d'aquest IDE, que s'accedeix a través de l'URL.



IL·LUSTRACIÓ 29: GRAPHQL IDE PLAYGROUND

No només permet cridar als mètodes públics, també exposa l'esquema GraphQL i la documentació generada automàticament per GraphQL – també documentada en l'apartat 6.3.1-.

7.2.3 Servei d'allotjament d'imatges

En aquest projecte s'ha ficat molta èmfasi a l'experiència d'usuari. Tot i que la major part d'aquesta responsabilitat cau en el frontend, el backend també afecta a aquesta. Durant el desenvolupament s'ha detectat que les imatges mostrades en el frontend tardaven massa en carregar-se, empitjorant així l'experiència de l'usuari. Per aquesta raó s'ha fet servir un servei d'allotjament d'imatges: Cloudinary [36]. Convé ressaltar que aquest canvi de disseny durant la fase d'implementació ha sigut possible gràcies a la metodologia àgil aplicada, ja que ha permet rectificar ràpidament en el següent Sprint.

En primer lloc, l'ús d'un servei com aquest permet una **major optimització i disponibilitat de les imatges**, produint així temps d'espera menor en el client. Addicionalment, Cloudinary té una llibreria de NodeJS, acompanyada d'una bona documentació, facilitant així la integració d'aquest servei en el backend. Finalment, aquesta eina disposa de diferents plans de preus que, afortunadament, el gratuït és més que suficient per l'estat actual del projecte.

7.2.4 Algorisme de paginació

En l'apartat 7.1.3.3 s'ha comentat els avantatges de la paginació a l'hora de demanar els anuncis d'una certa categoria. En aquest apartat es valorarà l'algorisme fet servir respecte als altres que intenten solucionar el mateix problema, justificant així el seu ús. A més, s'explicarà detalladament els passos de l'algorisme.

7.2.4.1 Anàlisi d'alternatives

Suposem que treballem amb el mètode següent per demanar els anuncis de la categoria gossos: `ads(category: DOGS)`, que ens retorna la llista completa de tots els anuncis d'aquesta categoria. En aquests moments aquest mètode només ofereix un paràmetre per especificar la categoria.

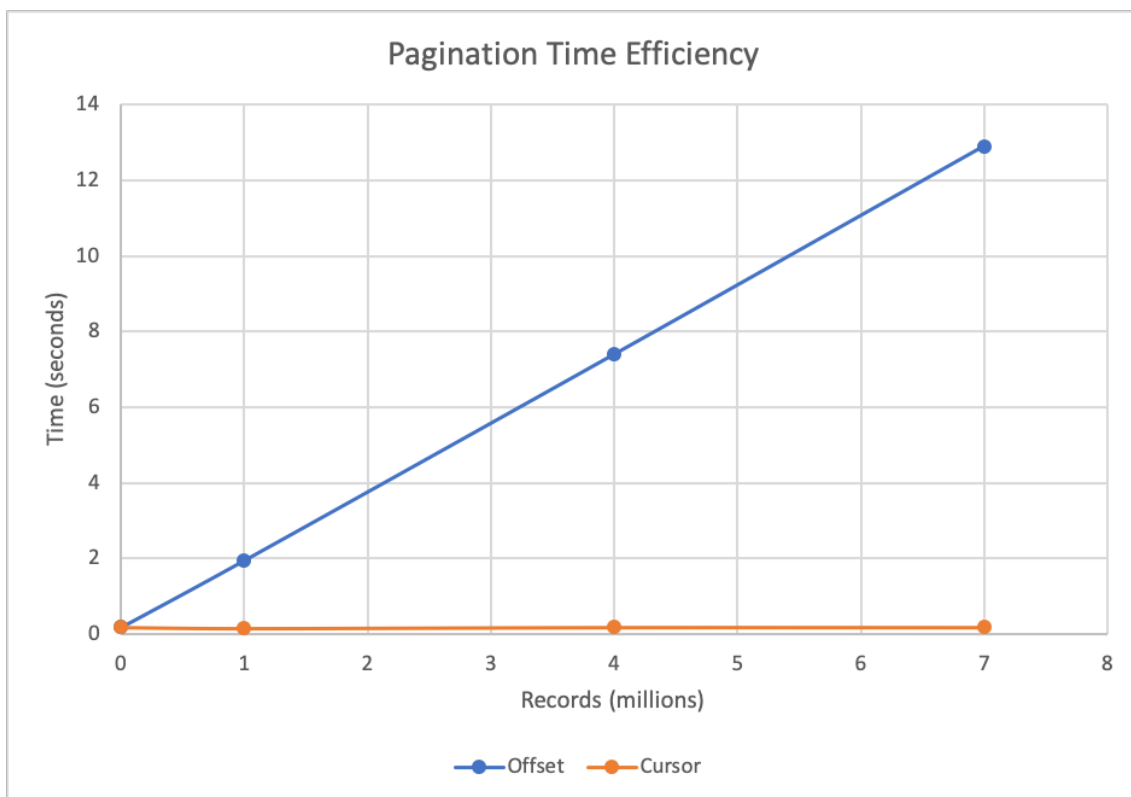
Ara, hi ha diferents maneres d'implementar la paginació en aquest mètode:

1. **Paginació basada en *offsets***: Podríem fer `ads(category: DOGS, first: 10, offset: 2)` per obtenir els 10 primers anuncis – *first: 10*- després dels 2 primers – *offset: 2*-.
2. **Paginació basada en *IDs***: Una altre opció seria fer `ads(category: DOGS, first: 10, after: $adId)` per obtenir els 10 anuncis – *first: 10*- després de l'anunci identifica't per *\$adId* demanat a la darrera petició – *after: \$adId*-.
3. **Paginació basada en *cursors***: O per concloure, es podria fer `ads(category: DOGS, first: 10, after: $adCursor)` per obtenir els 10 anuncis – *first: 10*- després del cursor que identifica a l'últim element demanat a la darrera petició – *after: \$adCursor*-.

Usant una paginació basada en *offsets* – opció 1- comporta certes restriccions, com ara que les dades han de ser estàtiques. De manera que si no ho són i s’implementa aquesta solució poden passar dues opcions: o retornar anuncis duplicats o saltar-se alguns anuncis. Aquest comportament no és el desitjat i pot portar a futures complicacions.

A més, l’*offset* només indica el numero de d’entrades que la base de dades ha de saltar per arribar als anuncis en concret. Això es tradueix a que la base de dades no només retornarà els anuncis demanats, també està escanejant totes les dades anteriors a aquests anuncis.

La Il·lustració 30 exposa una gràfica que mostra el temps que tarda la paginació basada en offset i la paginació basada en cursors sobre una base de dades de fins a 7 milions d’entrades.



IL·LUSTRACIÓ 30: GRÀFICA COMPARATIVA ENTRE LA PAGINACIÓ BASADA EN CURSORS I LA BASADA EN OFFSET

A partir d’aquesta il·lustració es pot observar que quan l’usuari demani anuncis amb un *offset* de 7 milions l’usuari haurà d’esperar 13 segons, en canvi amb la implementació de la paginació basada en cursors la mateixa petició trigaria 0.18 segons. Així mateix, es descarta l’ús de la paginació basada en *offsets*.

Tot i que la segona opció és perfectament viable, les dues primeres opcions es poden implementar fent servir l’última opció – convertint el cursor amb l’*ID* o l’*offset*-. Addicionalment l’ús de cursors incrementa la flexibilitat del sistema en el cas de que s’hagi de canviar el model de paginació en un

futur. Convé ressaltar que els cursors són cadenes de caràcters opaques, que no s'ha de confiar en el seu format – ja que pot canviar amb el temps-, i es recomana codificar-lo en base64²².

7.2.4.2 Algorisme de paginació basat en cursors

Un cop seleccionat l'algorisme a implementar es procedeix a aplicar-lo. Tot i que seguidament s'explicaran els passos a seguir en l'algorisme basat en cursors, es pot trobar més informació en l'especificació formal [37] feta per Relay Project [38]. Des de la web oficial de GraphQL [13] recomanen seguir aquesta documentació a l'hora d'implementar una GraphQL API amb paginació.

7.2.4.2.1 Paràmetres

Primer de tot comentarem els paràmetres que pot acceptar aquest algorisme. A l'hora de paginar una llista d'entrades, existeixen tres opcions: paginació cap endavant, paginació cap enrere, o ambdues.

Per una paginació cap endavant es necessiten els següents dos paràmetres: *first* i *after*. En canvi, per una paginació cap enrere es necessiten els paràmetres *last* i *before*. On:

- Els paràmetres *first* i *last* són nombres naturals més grans que 0.
- Els paràmetres *after* i *before* són cursors que senyalen al darrer element de la pàgina anterior. En el cas de ser la primera pàgina demanada aquests valors seran nulls, és a dir buits.

7.2.4.2.2 Procés

Per determinar quines entrades retornar, els cursors *before* i *after* filtraran aquestes, i darrerament s'avaluarà *first* i *last* per tallar la resta. A continuació es detallen els mètodes de manera formal:

entradesARetornar(totesEntrades, before, after, first, last) :

1. Inicialitzar la variable *entrades* al resultat de cridar el mètode *aplicaCursorsALesEntrades(totesEntrades, before, after)*.
2. Si *first* no és buit:
 - a. Si *first* és més petit que 0:
 - i. Llença un error.
 - b. Si *entrades* té una llargada superior a *first*:
 - i. Tallar les ultimes *entrades* fins tenir una llargada igual a *first*.
3. Si *last* no és buit:
 - a. Si *last* és més petit que 0:

²² Grup d'esquemes de codificació de binari a text que representen dades binàries mitjançant una cadena ASCII.

- i. Llença un error
 - b. Si *entrades* té una llargada superior a *last*:
 - i. Tallar les primeres *entrades* fins a tenir una llargada igual a *last*.
4. Retorna *entrades*.

aplicaCursorsALesEntrades(totesEntrades, before, after) :

1. Inicialitzar *entrades* a *totesEntrades*.
2. Si *after* no és buit:
 - a. Inicialitzar *entradaAfter* a l'entrada la qual apunta el cursor *after*.
 - b. Si l'*entradaAfter* existeix:
 - i. Treure tots els elements davant de l'*entradaAfter*, incloent-hi aquesta.
3. Si *before* no és buit:
 - a. Inicialitzar *entradaBefore* a l'entrada la qual apunta el cursor *before*.
 - b. Si l'*entradaBefore* existeix:
 - i. Treure tots els elements de darrera de l'*entradaBefore*, incloent-hi aquesta.
4. Retornar *entrades*.

Aquests mètodes descrits són el nucli de l'algorisme. Addicionalment, es pot implementar més mètodes d'ajuda a fi d'exposar al client si hi ha més pàgines disponibles; d'aquesta manera no caldrà que el client efectuï una petició de més quan s'acabin aquestes.

7.2.5 Base de dades: MongoDB

La base de dades d'un software és un element important a comentar, ja que aquest component afecta de forma directa a l'evolució de la plataforma. Tot i que les bases de dades relacionals són més madures i disposen d'estàndards millor definits, les bases de dades no relacionals aporten una major versatilitat. Aquest avantatge és clau a l'hora de desenvolupar un projecte nou com aquest, ja que té alta probabilitat de sofrir canvis. A més, una base de dades no relacional és més sostenible, per tant permet de forma fàcil un creixement sobtós de les dades.

Per aquestes raons, s'ha escollit MongoDB [12] com a base de dades de la plataforma PetsWorld. Encara que MongoDB disposa d'una llibreria oficial per la seva gestió en NodeJS, s'ha fet ús d'una llibreria no oficial – *mongoose* [27]-; ja que disposa de més comunitat i menys complexitat. Per exemple, *mongoose* proporciona una classe repositori a partir de l'esquema de la base de dades de MongoDB.

7.2.6 Desplegament

Finalment, l'últim pas a realitzar en el desenvolupament del backend és el desplegament. En aquest apartat s'exposen els passos seguits per desplegar el *software* del backend en un servidor públic. D'aquesta manera es pot accedir al servei des de qualsevol punt geogràfic.

Prèviament és necessari disposar d'un servidor públic, que en el meu cas és un servidor amb el sistema operatiu Ubuntu LTS 18.04, localitzat físicament a Londres, contractat a l'empresa DigitalOcean [39]. Anteriorment aquest servidor ja disposava d'un firewall²³ configurat per permetre connexions al port 80 amb el protocol TCP. Aquesta configuració permetrà a l'aplicació mòbil comunicar-se amb el backend a través de HTTP i WS. A més, es va adquirir i configurar un domini DNS per accedir al servidor a través de l'URL raulmabe.dev, facilitant així l'accés al servidor.

Un cop disposat d'un servidor, s'ha procedit a automatitzar el desplegament a través de *Git Hooks*. *Git Hooks* són *scripts*²⁴ que s'executen automàticament cada cop que un esdeveniment particular ocorre en un repositori git.

Seguidament, es realitzen els següents passos per la seva configuració:

1. Inicialitzar un repositori git en el servidor amb un *GitHook post-receive – script* que s'executa després de rebre un esdeveniment-.
2. Aquest *script* ha d'encarregar-se d'iniciar l'aplicació en concret, en aquest cas el backend de la plataforma.
3. Assegurar els permisos de l'script de forma que sigui executable.
4. Per acabar, només s'ha d'afegir un nou *remote*²⁵ al repositori git que teníem fins ara en la màquina local, sent aquest *remote* el repositori inicialitzat en el pas 1.

Ara, cada cop que es vulgui actualitzar el backend públic del servidor només caldrà fer ús de la comanda "*git push <remote_name> master*" des de la màquina local.

8 Validació i proves

Un cop implementat el sistema, és el moment de validar-lo. Per validar-lo es comprovarà que tots els requeriments definits prèviament a l'apartat 5 se satisfan. Tot i que s'han implementat tests durant el desenvolupament, aquests tests no verifiquen el correcte comportament dels requisits funcionals, és per aquest motiu que s'ha usat una validació manual per validar aquests.

²³ Part d'un sistema informàtic dissenyada per denegar l'accés no autoritzat.

²⁴ Document executable que conté una sèrie d'instruccions.

²⁵ Un repositori comú on guardar els canvis del codi, com ara els servidors de Github.

Respecte als tests automàtics, s'han implementat diferents tests unitaris per verificar el correcte funcionament d'uns mètodes en concret. Però, donada la dificultat de la creació de tests de widgets – tests d'UI-, com de tests d'integració en aplicacions mòbil s'ha cregut oportú dedicar aquest temps més endavant, un cop implementat i validat el mínim producte viable.

Ara bé, abans d'actualitzar en un futur el *software* implementat seria ideal disposar de suficients tests de widgets i unitaris per disposar un *code coverage*²⁶ de més del 50%; i a més, suficients tests d'integració per cobrir tots els casos d'ús del projecte.

Pel que fa a la validació manual, es provaran diferents escenaris per cada cas d'ús a fi de determinar el seu correcte funcionament. Convé ressaltar que els tests manuals aporten un grau més de realisme, simulant amb més precisió un escenari autèntic on intervenen persones verídiques.

Per acabar, per assegurar la validació de cada requisit no funcional s'avaluaran els criteris de satisfacció de cadascun.

8.1 Validació manual dels requisits funcionals

Per cada cas d'ús es realitzaran varies probes per tal de simular els diferents escenaris. Per tal de documentar aquestes probes s'exposarà el resultat en forma de taula, amb els següents camps:

- **Escenari:** descriu la prova i les circumstàncies en què es realitza.
- **Resultat:** resultat de la prova: **èxit** o **fracàs**.
- **Observacions:** comentaris sobre la prova en qüestió.

R001 – Registrar-se		
Escenari	Resultat	Observació
L'actor omple el formulari de registre amb dades correctes.	Èxit	L'usuari ha sigut registrat satisfactòriament en el sistema i ha sigut autoritzat a realitzar funcionalitats restringides de l'app.
L'actor omple el formulari de registre amb dades incorrectes.	Èxit	El sistema no deixa registrar-se a un usuari amb dades invàlides, informant a l'usuari de que aquestes són incorrectes.

TAULA 32: VALIDACIÓ DE LA FUNCIONALITAT REGISTRAR-SE

²⁶ Mesura que s'utilitza per descriure el grau en què s'executa el codi font d'un programa quan s'executa un conjunt de tests.

R002 – Iniciar sessió		
Escenari	Resultat	Observació
L'actor omple el formulari d'iniciar sessió amb dades correctes.	Èxit	L'usuari ha sigut iniciat sessió satisfactòriament en el sistema i ha sigut autoritzat a realitzar funcionalitats restringides de l'app.
L'actor omple el formulari d'iniciar sessió amb dades incorrectes.	Èxit	El sistema no deixa iniciar sessió a un usuari amb dades invàlides.

TAULA 33: VALIDACIÓ DE LA FUNCIONALITAT INICI DE SESSIÓ

R003 – Modificar perfil		
Escenari	Resultat	Observació
L'actor modifica qualsevol dada del seu perfil de forma correcte.	Èxit	El perfil de l'actor ha sigut actualitzat amb els nous valors.

TAULA 34: VALIDACIÓ DE LA FUNCIONALITAT MODIFICAR PERFIL

R004 – Consultar perfil propi o extern		
Escenari	Resultat	Observació
L'actor consulta un perfil extern al seu.	Èxit	L'actor pot visualitzar correctament el perfil extern.
L'actor consulta el seu perfil propi.	Èxit	L'actor pot visualitzar correctament el perfil propi.

TAULA 35: VALIDACIÓ DE LA FUNCIONALITAT CONSULTAR PERFIL PROPI O EXTERN

R005 – Enviar un missatge a un altre usuari		
Escenari	Resultat	Observació
L'actor envia un missatge vàlid a un altre usuari correctament.	Èxit	El sistema ha registrat el missatge enviat per l'actor.
L'actor intenta enviar un missatge invàlid a un altre usuari correctament.	Èxit	El sistema no ha deixat enviar el missatge invàlid, i per tant no l'ha registrat.

TAULA 36: VALIDACIÓ DE LA FUNCIONALITAT ENVIAR UN MISSATGE A UN ALTRE USUARI

R006 – Consultar anuncis		
Escenari	Resultat	Observació
L'actor consulta els anuncis de totes les categories disponibles.	Èxit	L'app mostra els anuncis disponibles corresponents a cada categoria.

TAULA 37: VALIDACIÓ DE LA FUNCIONALITAT CONSULTAR ANUNCIS

R007 – Cercar anuncis per filtres		
Escenari	Resultat	Observació
L'actor ha aplicat diferents filtres a l'hora de cercar anuncis.	Èxit	L'app mostra els anuncis que compleixen els filtres especificats.

TAULA 38: VALIDACIÓ DE LA FUNCIONALITAT CERCAR ANUNCIS PER FILTRES

R008 – Cercar protectores per proximitat		
Escenari	Resultat	Observació
L'actor pot cercar les protectores més pròximes a la seva adreça.	Èxit	L'app mostra, de forma ordenada, les protectores més properes a l'adreça de l'actor.
L'actor, que no té una adreça configurada, intenta cercar les protectores més pròximes.	Èxit	L'app informa a l'usuari la necessitat d'introduir la adreça per fer servir aquesta funcionalitat.

TAULA 39: VALIDACIÓ DE LA FUNCIONALITAT CERCAR PROTECTORES PER PROXIMITAT

R009 – Crear, actualitzar o eliminar una valoració a un perfil extern		
Escenari	Resultat	Observació
L'actor crea una valoració a un perfil extern.	Èxit	El sistema registra la valoració correctament i la fa pública.
L'actor edita una valoració feta amb anterioritat.	Èxit	El sistema actualitza la valoració correctament.

L'actor elimina una valoració feta amb anterioritat.	Èxit	El sistema elimina la valoració correctament.
--	------	---

TAULA 40: VALIDACIÓ DE LA FUNCIONALITAT CREAR, ACTUALITZAR O ELIMINAR UNA VALORACIÓ A UN PERFIL EXTERN

R010 – Crear o eliminar anuncis de mascotes		
Escenari	Resultat	Observació
L'actor, sent protectora o particular, crea un anunci d'una mascota.	Èxit	El sistema processa correctament el nou anunci i el mostra.
L'actor, sent professional, intenta crear un anunci d'una mascota.	Èxit	El sistema no deixa l'opció a l'usuari, ja que els professionals no poden pujar anuncis de mascotes.
L'actor elimina un anunci propi de mascotes publicat anteriorment.	Èxit	El sistema elimina l'anunci correctament.

TAULA 41: VALIDACIÓ DE LA FUNCIONALITAT CREAR O ELIMINAR ANUNCIS DE MASCOTES

R011 – Crear o eliminar anuncis de serveis		
Escenari	Resultat	Observació
L'actor, sent professional o particular, crea un anunci d'un servei.	Èxit	El sistema processa correctament el nou anunci i el mostra.
L'actor, sent protectora, intenta crear un anunci d'un servei.	Èxit	El sistema no deixa l'opció a l'usuari, ja que les protectores no poden pujar anuncis de serveis.
L'actor elimina un anunci propi de serveis publicat anteriorment.	Èxit	El sistema elimina l'anunci correctament.

TAULA 42: VALIDACIÓ DE LA FUNCIONALITAT CREAR O ELIMINAR ANUNCIS DE SERVEIS

R012 – Crear o eliminar anuncis d'articles		
Escenari	Resultat	Observació
L'actor, sent professional, crea un anunci d'un article.	Èxit	El sistema processa correctament el nou anunci i el mostra.

L'actor, sent protectora o particular, intenta crear un anunci d'un article.	Èxit	El sistema no deixa l'opció a l'usuari, ja que les protectores i particulars no poden pujar anuncis de mascotes.
L'actor elimina un anunci propi d'articles publicat anteriorment.	Èxit	El sistema elimina l'anunci correctament.

TAULA 43: VALIDACIÓ DE LA FUNCIONALITAT CREAR O ELIMINAR ANUNCIS D'ARTICLES

R013 – Guardar anuncis a preferits		
Escenari	Resultat	Observació
L'actor guarda un anunci a preferits.	Èxit	El sistema processa la petició i guarda l'anunci en concret a preferits de l'actor.

TAULA 44: VALIDACIÓ DE LA FUNCIONALITAT GUARDAR ANUNCIS A PREFERITS

R014 – Eliminar anuncis de preferits		
Escenari	Resultat	Observació
L'actor elimina un anunci de preferits.	Èxit	El sistema processa l'eliminació de preferits correctament.

TAULA 45: VALIDACIÓ DE LA FUNCIONALITAT ELIMINAR ANUNCIS DE PREFERITS

8.2 Estratègia de proves dels requisits no funcionals

Com s'ha comentat anteriorment, per comprovar la validació dels requisits no funcionals s'avaluaran els criteris de satisfacció de cadascun d'ells. A causa d'això s'ha fet ús de les següents diferents metodologies:

- **Enquesta als usuaris:** s'ha compartit una enquesta de Google entre 10 usuaris que han fet servir l'aplicació per saber l'opinió 'aquests respecte els requisits R017, R018 i R019.
- **Comprovació visual:** per comprovar alguns requisits en concret, com ara el R021, no és necessari interaccionar amb l'aplicació, però visualitzar les vistes d'aquesta, o visualitzar les dades encriptades de la base de dades, entre d'altres. Per aquests requisits s'usarà una comprovació visual.
- **Comprovació manual:** Per la resta de requisits, es farà una comprovació manual com s'ha fet anteriorment pels requisits funcionals.

8.3 Validació dels requisits no funcionals

De la mateixa manera que s'ha documentat les validacions de l'apartat 8.1, s'exposaran en forma de taula els següents atributs:

- **Descripció:** descripció del requisit no funcional.
- **Criteri de satisfacció:** condició que s'ha de complir per classificar el requisit com a satisfet.
- **Resultat:** resultat de la prova: **èxit** o **fracàs**.
- **Observacions:** comentaris sobre la prova en qüestió.

R015 - Descripció		La vista ofereix diferents temes per diferents gustos visuals.	
Criteri d'acceptació	Resultat	Observacions	
L'usuari és capaç d'aplicar el tema fosc de l'aplicació.	Èxit	L'aplicació ha canviat d'aparença i ha aplicat el tema fosc.	
L'usuari és capaç d'aplicar el tema clar de l'aplicació.	Èxit	L'aplicació ha canviat d'aparença i ha aplicat el tema clar.	

IL·LUSTRACIÓ 31: VALIDACIÓ DEL REQUISIT R015

R016 - Descripció		El disseny de la interfície resulta atractiu per a l'usuari	
Criteri d'acceptació	Resultat	Observacions	
Es realitzarà una enquesta als usuaris que hagin utilitzat el sistema més d'un cop, i el 80% o més confirmarà la satisfacció del requisit.	Èxit	El 100% dels usuaris enquestats han contestat favorablement.	

IL·LUSTRACIÓ 32: VALIDACIÓ DEL REQUISIT R016

R017 - Descripció		El disseny de la interfície resulta senzill d'utilitzar per a l'usuari	
Criteri d'acceptació	Resultat	Observacions	
Es realitzarà una enquesta als usuaris que hagin utilitzat el sistema més d'un cop, i el 80% o més confirmarà la satisfacció del requisit.	Èxit	El 100% dels usuaris enquestats han contestat favorablement.	

IL·LUSTRACIÓ 33: VALIDACIÓ DEL REQUISIT R017

R018 - Descripció		El sistema té un llenguatge comprensible per a un usuari normal
criteri d'acceptació	Resultat	Observacions
<p>S'utilitzarà un llenguatge clar i concret. A més, es proporcionarà l'habilitat de canviar l'idioma de l'aplicació.</p> <p>Durant l'enquesta als usuaris, es valorarà que aquests entenguin bé el contingut de l'aplicació.</p>	Èxit	<p>S'ha utilitzat un llenguatge clar i concret. A més s'ha proporcionat l'opció de canviar d'idioma entre anglès, castellà i català.</p> <p>Els usuaris han demostrat una perfecte comprensió del llenguatge de l'<i>app</i> durant l'enquesta.</p>

IL·LUSTRACIÓ 34: VALIDACIÓ DEL REQUISIT R018

R019 - Descripció		El sistema utilitza icones fàcils d'identificar
criteri d'acceptació	Resultat	Observacions
<p>S'intentarà, quan sigui possible, utilitzar icones ja establerts pel framework Flutter, adherits al Material Design.</p>	Èxit	Més del 80% dels icones utilitzats pertanyen a Material Design.

IL·LUSTRACIÓ 35: VALIDACIÓ DEL REQUISIT R019

R020 - Descripció		Els textos que mostra l'aplicació són totalment llegibles
criteri d'acceptació	Resultat	Observacions
<p>Es seguirà les pautes establertes per Material Design per assegurar un correcte ús dels colors per a textos.</p>	Èxit	<p>S'han seguit les pautes establertes per Material Design [40] a l'hora d'escollir els colors de cada component.</p>

IL·LUSTRACIÓ 36: VALIDACIÓ DEL REQUISIT R020

R021 - Descripció		El sistema és escalable
criteri d'acceptació	Resultat	Observacions
<p>S'utilitzen eines, com MongoDB, tenint en compte l'escalabilitat d'aquestes</p>	Èxit	<p>Les eines han estat seleccionades amb cura per tenir en compte l'escalabilitat del sistema. També s'ha tingut en compte a l'hora de dissenyar l'arquitectura lògica del sistema.</p>

IL·LUSTRACIÓ 37: VALIDACIÓ DEL REQUISIT R021

R022 - Descripció	El sistema guarda les preferències de l'usuari encara que s'hagi sortit de l'aplicació	
Criteri d'acceptació	Resultat	Observacions
Es guardaran les dades de les preferències en local, i al entrar de nou a l'aplicació, el sistema recuperarà aquestes dades.	Èxit	Les preferències de l'usuari com l'idioma i l'aparença de l'aplicació són guardades en local; i quan l'usuari torna a obrir l'aplicació aquestes es carreguen i s'apliquen satisfactòriament.

IL·LUSTRACIÓ 38: VALIDACIÓ DEL REQUISIT R022

R023 - Descripció	El sistema ha de guardar la informació sensible dels usuaris de forma encriptada o hashejada	
Criteri d'acceptació	Resultat	Observacions
S'encriptarà o es hashejarà tota informació sensible	Èxit	Les contrasenyes de cada usuari són encriptades a través de la funció bcrypt [36]

IL·LUSTRACIÓ 39: VALIDACIÓ DEL REQUISIT R023

R024 - Descripció	Totes les dades de caràcter personal han de ser tractades complint la LOPD Espanyola (Llei orgànica de protecció de dades 15/1999)	
Criteri d'acceptació	Resultat	Observacions
S'encriptarà o es hashejarà tota informació sensible	Èxit	Les contrasenyes de cada usuari són encriptades a través de la funció bcrypt [36]

IL·LUSTRACIÓ 40: VALIDACIÓ DEL REQUISIT R024

9 Resultats de la gestió del projecte

Un cop acabada la fase d'implementació i la fase de validació és hora de detallar els resultats de la gestió del projecte. A continuació es comenten els resultats de la metodologia aplicada, les desviacions respecte a la planificació inicial i la planificació econòmica. I per acabar, s'analitza la sostenibilitat del projecte.

9.1 Resultats a nivell de la metodologia

9.1.1 Mètodes de treball

D'entrada, gràcies a la metodologia escollida inicialment – metodologia Agile- ha permès un desenvolupament dinàmic, on per cada limitació afrontada s'ha pogut respondre de manera ràpida i efectiva en la següent iteració. En canvi, desenvolupar aquest mateix projecte amb una metodologia tradicional hauria afectat severament la planificació inicial en qüestió de temps i costos.

Respecte a la metodologia Kanban, l'ús del taulell Kanban ha estat extremadament útil a l'hora de definir i implementar les tasques de cada *Sprint*. A més, gràcies al taulell s'ha classificat de forma visual cada tasca, millorant així l'eficiència i la productivitat del desenvolupament. Addicionalment, el taulell aporta flexibilitat i predictibilitat a l'hora de definir una data aproximada per finalitzar cada iteració.

Quant a la metodologia Scrum, s'ha fet un ús correcte del concepte d'*Sprints*, ja que s'ha aprofitat satisfactòriament els seus avantatges. Ha aportat claredat a l'hora de planificar, velocitat a l'hora d'implementar, i facilitat a l'hora d'iterar. Així mateix, gràcies a les reunions retrospectives l'autor del projecte ha dedicat temps a l'autocrítica, millorant així la tècnica aplicada a cada *Sprint*.

Per concloure, en conseqüència al resultat de la implementació i validació del projecte, es considera un èxit la metodologia triada i aplicada, ja que ha ajudat significativament la planificació i la implementació d'aquest.

9.1.2 Eines de seguiment

Totes les eines de seguiment esmentades inicialment en l'apartat X han aportat un gran valor i han estat utilitzades correctament en la implementació del projecte.

En primer lloc, l'eina més essencial i important a l'hora d'implementar un software és el sistema de control de versions de codi: Git. Aquesta eina ha aportat gran valor en termes de productivitat, podent controlar les diferents versions del codi i podent provar noves funcionalitats sense haver de perdre el codi anterior. Convé ressaltar que aquesta eina és totalment essencial en entorns de producció, ja que dóna una increïble seguretat tenint en compte que permet rectificar versions errònies.

En segon lloc s'ha utilitzat GitKraken a l'hora de fer servir Git. Com s'ha comentat en l'apartat X, GitKraken és una interfície gràfica per usar Git. Vist que Git és un sistema bastant complex, aquesta interfície aporta una gran simplicitat i usabilitat. En altres paraules, gràcies a l'ús d'aquesta eina l'autor del projecte no ha començat cap error usant Git, reduint així els possibles riscos.

En tercer lloc s'ha fe ús de Github. Un gestor de repositoris git és essencial per a l'ús d'aquesta tecnologia. Això ha permès compartir codi entre diferents dispositius, facilitant i agilitzant el procés de teletreball amb la situació del COVID-19.

En últim terme, l'ús de GitKraken Glo Boards ha permès aplicar perfectament la metodologia Kanban al projecte. A més, aquesta eina disposa d'una integració total al software GitKraken, aportant proximitat a l'entorn de treball.

9.1.3 Mètode de validació

Tot i que el mètode de validació empleat en el projecte ha sigut més que suficient, idealment s'hauria d'haver utilitzat més el mètode de validació automàtica a través dels tests d'integració. L'únic mètode automàtic utilitzat han estat els unit test, que tot i que són útils per validar mètodes, no són adequats a l'hora de validar les funcionalitats i els casos d'ús de la plataforma. Llavors, com s'ha comentat en l'apartat 8, realitzar suficients tests d'integració per validar tots els casos d'ús de l'aplicació i, a més, obtenir un *code coverage* de més del 50% seria idoni. Malgrat que no serà documentat, aquesta tasca és la següent a fer, abans d'ampliar els casos d'ús del sistema.

Per altra banda, quant a la documentació i memòria d'aquest treball, la tutora – Prof. Cristina Gómez Seoane- ha servit de gran ajuda a l'hora de puntualitzar errors o falta de profunditat en certs apartats. A més, qualsevol dubte que ha sorgit ha contestat ràpidament i eficientment.

9.2 Resultats a nivell de la planificació

La planificació esmentada en l'apartat 4.2 ha estat essencial per poder desenvolupar un projecte de forma estructurada i eficient. Addicionalment, es va detallar una estimació per cada fase del projecte i cada *Sprint* basant-se en l'experiència de l'autor implementant i realitzant treballs previs. Ara bé, ja que no deixen de ser estimacions, la planificació final ha patit certes desviacions. En la següent taula es pot visualitzar la planificació final.

Codi	Tasca	Hores estimades	Hores reals	Data inici	Data final
T001	Planificació i gestió del projecte	75	75	17/02	16/03
T002	Desenvolupament de l'aplicació	480	314	17/03	30/07
T003	Anàlisi de requisits	30	15	17/03	31/03
T004	Especificació i disseny	30	26	1/04	15/04
T005	Implementació	420	273	16/04	30/08
T014	Documentació i defensa	50	60	17/03	16/10
T015	Redacció de la memòria i reunions de seguiment	40	50		
T016	Preparació i presentació oral	10	10		
TOTAL		605	444	17/02	23/10

TAULA 46: ETAPES DEL DESENVOLUPAMENT

Planificació i gestió del projecte

Aquesta fase no ha patit cap canvi respecte a l'estimada inicialment.

Desenvolupament de l'aplicació

Quant a la fase de desenvolupament de l'aplicació, s'ha endarrerit 30 dies respecte a l'estimació inicial. Aquesta desviació ha sigut producte de la incerta i excepcional situació mundial sobre la pandèmia del COVID-19.

Afortunadament, aquest retard no ha afectat la realització del projecte ni a la qualitat d'aquest, ja que l'autor va preveure aquesta possibilitat. El pla de mitigació va ser optar pel segon torn de lectura, obtenint així suficient temps extra per a superar les adversitats produïdes.

Adicionalment, aquesta fase ha patit una desviació més, respecte a les hores estimades. Les hores finals mostren aproximadament un terç de desviació respecte de les estimades – 166 hores menys-, fet que no sorprèn, ja que el pla de mitigació principal va ser estimar les hores de cada *Sprint* a l'alta, a causa de la poca experiència en certes eines utilitzades en aquest projecte.

Documentació i defensa

Respecte a la fase final, de documentació i defensa, les hores estimades han reflectit prou bé les hores finals, patint només una desviació de 10 hores a l'hora de documentar. Quant a la preparació de la memòria, s'ha mantingut l'estimació inicial, ja que aquesta tasca s'iniciarà justament després de finalitzar la memòria.

9.3 Resultats a nivell de gestió econòmica

Tenint en compte les desviacions esmentades anteriorment, l'estimació del cost del projecte que es mostra en l'apartat 4.3.1.5 ha patit canvis. És per això que s'ha tornat a calcular el cost del projecte, basant-se així en les hores reals del desenvolupament del projecte. En la següent taula – Taula 47– es mostra com queden els costos directes finals per a cada rol, i pel total.

Rol	Remuneració (en €/h)	Hores reals totals	Cost estimat (en €)	Cost real (en €)
Cap de projecte	37,00	130	4.625,00	4.810,00
Analista programador	15,70	15	471,00	251,20
Programador sènior	16,38	15,6	294,75	255,53
Programador júnior	9,00	273	3.780,00	2.457,00
Dissenyador gràfic	9,38	10,4	112,50	97,55
TOTAL			9.283,25 €	7.871,28 €

TAULA 47: COSTOS DIRECTES FINALS PER ROLS

Tal com es mostra, els costos directes han disminuït 1.411,97 € a causa de les desviacions. A causa d'això, ni la partida de contingències ni la partida d'imprevistos han estat necessàries. Resultant així amb un **cost total del projecte de 7.871,28 €**.

9.4 Sostenibilitat

A l'hora de realitzar aquest apartat, s'ha demanat a l'autor del treball realitzar una enquesta sobre la sostenibilitat de projectes TIC. Un cop realitzada l'enquesta, l'estudiant i autor del treball de final de grau ha quedat sorprès respecte al gran nombre de factors que involucren la sostenibilitat d'un projecte de software.

Tot i que l'estudiant considera que és bastant conscient sobre l'impacte social i econòmic de projectes relacionats amb les TIC, ja que ha adquirit suficients coneixements al llarg dels seus estudis universitaris sobre el desenvolupament de software; s'ha adonat sobre el poc conscient que n'és sobre l'impacte ambiental.

Així mateix, l'autor assumeix el fet de no haver comprès tots els conceptes mencionats en l'enquesta, i per tant, una manca de coneixements en el compromís que ha de mostrar l'Enginyeria Informàtica

amb el medi ambient. És per això que s'ha arribat a la conclusió que ha d'estudiar més aquests aspectes, ja que són igual o més importants que els altres dos.

Finalment, si en algun moment l'estudiant assoleix una posició de responsabilitat en un projecte d'informàtica, no permetrà cap presa de decisions sense haver estudiat abans l'impacte ambiental d'aquest.

9.4.1 Dimensió econòmica

Es creu que la realització dels pressupostos ha sigut coherent amb la realitat, fent així un ús poc excessiu de recursos humans i materials.

Així mateix, gràcies a l'ús de tecnologies com Flutter poden desenvolupar una aplicació multiplataforma per a Android i iOS amb un únic equip de desenvolupament, reduint així els costos de recursos humans a quasi la meitat.

Tanmateix, la reutilització de recursos materials com l'ús de l'ordinador particular de l'autor s'aconsegueix disminuir al màxim el cost de materials.

Addicionalment, al ser un projecte educatiu els costos del projecte han estat mínims, resultant així en un preu altament competitiu. S'estima per tant que l'impacte econòmic en producció té una nota de 10/10.

D'altra banda, aquest producte únicament requereix un servidor per mantenir-lo viu. Aquest servidor té un cost de 6 € al mes. En cas d'un augment d'usuaris es podria ampliar el servidor perfectament. D'aquesta manera s'estima que l'impacte econòmic en vida útil d'aquest projecte serà també de 10/10.

Quant als riscos, existeixen diferents esdeveniments que poden afectar a la viabilitat del projecte, com ara la falta d'usuaris. Així mateix, aquest projecte requereix un pla de *marketing* per poder ser altament viable en termes econòmics. Per concloure, es dona a aquest punt una nota de 4/10.

9.4.2 Dimensió ambiental

En el desenvolupament d'aquest projecte s'ha procurat utilitzar el mínim nombre de recursos possibles. En aquest procés només es requereix energia elèctrica pel portàtil de l'autor, i més endavant en el procés de manteniment d'aquest projecte, per mantenir encès un únic servidor que manté el back-end en marxa.

Pel que fa als materials utilitzats, com s'ha mencionat en l'apartat anterior, s'ha reutilitzat un ordinador particular, ja que no havia necessitat d'adquirir un de nou. D'aquesta manera no només

s'aconsegueix reduir al màxim el pressupost, sinó també la petjada ambiental del treball. Així mateix, la nota del projecte en aquest apartat serà de 9/10.

Durant la vida útil del projecte, el sistema requerirà un mínim ús de la bateria del dispositiu de l'usuari. Addicionalment, el servidor contractat per a mantenir el sistema viu és subcontractat, per tant es pot confiar en l'optimització de recursos de l'empresa en qüestió per maximitzar el guany d'aquesta. A aquesta part se li dóna una nota de 10/10.

Per acabar, els riscos que fiquen en perill la dimensió ambiental del sistema són altament improbables, resultant així en una nota de 10/10.

9.4.3 Dimensió social

Aquest projecte està enfocat principalment en aconseguir un impacte social positiu. Com s'ha descrit anteriorment, l'objectiu del treball és facilitar al màxim l'adopció de mascotes les quals no tenen llar. Així llavors, la realització d'aquesta plataforma ofereix un gran impacte social, no només per les famílies, que a través de l'aplicació podran adoptar un nou membre; sinó també per salvar el màxim número de vides, sense diferenciar espècies.

D'altra banda, tot i que menys important, és rellevant mencionar que la realització d'aquest TFG aporta una gran experiència i una adquisició significativa de coneixements a nivell social a l'autor, sent així el TFG una gran oportunitat de realitzar un projecte tan complet.

El problema que intenta solucionar aquest projecte ha servit com a oportunitat per a investigar més profundament en l'estat actual de les adopcions de mascotes sense llar. A més, s'ha conegut nous contactes – veterinaris, voluntaris d'associacions animalistes, i treballadors de protectores- que han aportat a l'autor un punt de vista més crític i verídic. En aquest apartat se li dóna una nota de 10/10.

Com s'ha mencionat anteriorment i a l'apartat 9, aquest projecte va directament adreçat a la millora de vida dels animals sense llar d'aquest país. Així mateix, l'èxit d'aquest projecte equivaldria a un augment de mascotes amb llar, millorant així no només la vida d'aquests animals, sinó també de tots els membres de les famílies adoptants. La nota en aquest aspecte és de 10/10.

Un risc que s'ha de tenir clarament en compte és permetre compravenda d'animals a través de l'aplicació. Abans de ser publicada com a aplicació real, s'afegirà l'opció de denunciar anuncis on els animals siguin tractats com a objecte, o hi hagi senyals clares de què l'animal és maltractat. Per aquest motiu s'intenta donar veracitat als usuaris de l'aplicació a través de valoracions de perfils. En aquest aspecte la nota és de 5/10

Després d'aquestes reflexions i valoracions, la taula següent mostra la matriu de sostenibilitat del TFG, recollint les notes mencionades anteriorment.

	PPP	Vida útil	Riscos	Total
<i>Dimensió econòmica</i>	10	10	4	24/30
<i>Dimensió ambiental</i>	9	10	10	29/30
<i>Dimensió social</i>	10	10	5	25/30
<i>Total</i>	29/30	30/30	19/30	78/90

TAULA 48: MATRIU DE SOSTENIBILITAT DEL TFG

Tot i que hi ha aspectes en els quals podria millorar, es considera que el projecte ha tingut en compte altres aspectes importants, resultant així en una bona nota de sostenibilitat.

10 Conclusions

10.1 Limitacions i dificultats

Durant la realització del treball l'autor ha patit varies limitacions i dificultats, on cada una d'elles ha causat petites desviacions respecte a la planificació inicial.

La limitació més excepcional ha estat la convocació de l'estat d'alarma de tot el país en els primers mesos del desenvolupament. Aquest estat ha negat a l'autor a moure's entre el seu lloc de residència d'estudiant, i el lloc de residència habitual. Addicionalment, aquest estat d'alarma ha empitjorat significativament la salut mental de la majoria de persones, ja que no estàvem acostumats a quedar-nos 24 hores i 7 dies tancats en una residència.

Una altra limitació ha sorgit per part de llibreries relacionades amb GraphQL, les quals no estan prou ben mantingues. La causa d'això és probable que sigui per la falta de temps dels desenvolupadors implicats o simplement que la tecnologia és bastant nova.

D'altra banda, navegadors com Google Chrome i Firefox han estat actualitzats recentment, i han denegat el tràfic a servidors amb un domini .dev a través de HTTP, redirigint aquest tràfic a través de HTTPS [41]. Aquesta jugada per part de Google no va ser prevista en cap moment, produint així certes dificultats a l'hora de desplegar el back-end en un servidor amb un domini .dev.

10.2 Integració de coneixements

Durant el desenvolupament d'aquest projecte s'han treballat diferents disciplines. Des del disseny de cada una de les interfícies, fins al desenvolupament, manteniment i desplegament de cada software necessari per al correcte funcionament del projecte.

Seguidament es llisten diferents assignatures del grau les quals el seu temari ha estat aplicat durant el desenvolupament d'aquest projecte.

Bases de Dades (BD) i Disseny de Bases de Dades (DBD)

Després de cursar aquestes dues assignatures l'autor ha adquirit els coneixements suficients per escollir amb consciència quin tipus de bases de dades requereix el projecte, i d'aquesta manera evitar futures possibles complicacions relacionades.

Concretament s'ha escollit una base de dades no relacional, com MongoDB per la seva escalabilitat, documentació, i comunitat.

Xarxes de Computadors (XC), Seguretat Informàtica (SI) i Administració de Sistemes Operatius (ASO)

Per tal de poder comunicar el frontend amb el backend ha sigut necessari treballar amb protocols, IPs²⁷, i servidors al núvol.

Per exemple, per la implementació d'un xat en temps real s'ha treballat amb el protocol WebSocket²⁸. A més, per la comunicació amb la resta de l'API s'ha utilitzat el protocol Hypertext Transfer Protocol²⁹ (HTTP). Paral·lelament, en el servidor públic on està corrent el backend s'ha hagut de configurar el domini raulmabe.dev³⁰ per poder accedir de forma més fàcil i directe al servidor. Tanmateix, per temes de seguretat s'ha hagut de configurar un firewall, indicant explícitament cada protocol i port permès.

Interacció i Disseny d'Interfícies (IDI)

L'experiència d'usuari és un punt clau en aquest projecte, ja que un dels requisits no funcionals és una bona experiència d'usuari dins l'aplicació. Gràcies als coneixements adquirits en IDI, l'autor ha sigut capaç d'aplicar principis de disseny d'interfícies en el projecte.

Estructura de Dades i algorismes (EDA)

²⁷ Adreça que identifica una interfície de xarxa

²⁸ Protocol usat per la comunicació entre dos sistemes informàtics en temps real

²⁹ Altra protocol usat per la comunicació entre dos sistemes informàtics, aquest en canvi no és en temps real

³⁰ Domini web, propietat de l'autor

L'ús d'estructures de dades en el projecte és extens (maps, llistes...). A més, s'han aplicat algorismes (com per exemple l'algorisme de paginació cursor-based) tenint en compte el cost d'aquests i de les diferents alternatives per a fer-lo el més eficient possible.

Arquitectura del Software (AS) i Projectes de Programació (PROP)

Gràcies als coneixements estudiats en Arquitectura del Software sobre els diferents tipus de patrons, l'ús de tests, i bones pràctiques, s'ha pogut dur a terme el desenvolupament del projecte sense grans inconvenients a l'hora d'implementar-lo.

A la vegada, l'assignatura PROP va ser una gran oportunitat per treballar de manera pràctica els coneixements d'AS; donant així més experiència a l'autor.

Aplicacions i Serveis Web (ASW)

Al llarg d'aquesta assignatura l'autor va poder provar diferents tecnologies usades en aplicacions i serveis web que han ajudat l'hora de triar quines tecnologies fer servir en aquest projecte.

Enginyeria de Requisits (ER)

Per la correcta realització d'un projecte és primordial conèixer els requisits d'aquest i la seva posterior validació. Per això, aplicar els coneixements adquirits durant l'assignatura d'Enginyeria de Requisits ha estat vital pel seu èxit.

Gestió de Projectes de Software (GPS)

Tanmateix, seguir una bona metodologia, i sobretot, saber triar quina és la més adequada en cada situació, també és primordial en qualsevol projecte de Software. Gràcies a aquesta assignatura s'ha fet un ús correcte de les eines disponibles per dur a terme de forma encertada la metodologia seleccionada.

Projecte d'Enginyeria del Software (PES)

En aquesta assignatura es va posar en pràctica tots els coneixements adquirits en l'especialització d'Enginyeria del Software per fer un projecte en equip. Gràcies a l'experiència, s'ha après dels errors comesos i s'han pogut evitar en l'actual projecte.

10.3 Justificació de les competències

CES1.1: Desenvolupar, mantenir i avaluar sistemes i serveis software complexos i/o crítics. [En profunditat]

L'objectiu d'aquest projecte ha estat desenvolupar un sistema complex, mantenir aquest a mesura que s'afegeixen noves funcionalitats, i avaluar diverses alternatives per cada una d'elles. El

desenvolupament del sistema no s'ha limitat a una aplicació monolítica, sinó que s'ha dividit en dos components principals: l'app multiplataforma i el servidor, comunicant-se entre ells a través d'un llenguatge de consultes. A més, s'ha fet ús de serveis externs i una base de dades no relacional, externa també. Així doncs, es considera que s'ha assolit aquesta competència en profunditat.

CES1.7: Controlar la qualitat i dissenyar proves en la producció de software. [Una mica]

El control de qualitat i la presència de proves per validar el correcte funcionament d'un projecte de software és essencial, és per això que s'ha dedicat un capítol complet per la validació d'aquest: 8 Validació i proves. En aquest capítol s'han validat els requisits funcionals i no funcionals, aplicant metodologies automàtiques i manuals. A més, es va realitzar una enquesta a 10 usuaris del sistema. Així doncs, es considera que s'ha assolit aquesta competència en el nivell indicat.

CES2.1: Definir i gestionar els requisits d'un sistema software. [Una mica]

Definir i gestionar els requisits d'un sistema de software és encara més essencial que el control de qualitat d'aquest. En el capítol Anàlisi de requisits i especificació es realitza un estudi detallat dels diferents requisits del sistema a desenvolupar. A més, els requisits no funcionals es relacionen amb els de Volere i es proporciona un criteri de satisfacció coherent per cada un d'ells. Pels requisits funcionals es determina l'escenari principal d'èxit i possibles escenaris secundaris. Així mateix, es considera que s'ha assolit aquesta competència satisfactòriament.

CES3.1: Desenvolupar serveis i aplicacions multimèdia. [Bastant]

El resultat final d'aquest projecte ofereix un servei per la gestió de dades d'una plataforma d'adopcions – ja que el servidor és públic -. A més, també ofereix una aplicació multimèdia: on a través d'imatges, textos, icones, i diàlegs s'aconsegueix transmetre un missatge més complet i entenedor a l'usuari final amb èxit. D'aquesta manera, es considera una competència totalment assolida en el nivell indicat.

10.4 Treball futur

Tot i que aquest treball està finalitzat, el projecte té molt de potencial i és bastant probable que s'afegeixin noves funcionalitats o es millorin les existents. Tot i que hi ha mil funcionalitats que es podrien afegir que millorarien la qualitat del producte i l'experiència d'usuari, seguidament es descriuen certes millores que podrien polir – encara més- l'aplicació mòbil i el sistema.

En primer lloc, és important saber l'opinió dels usuaris i donar l'opció a rebre feedback per part d'ells. És per això que la primera funcionalitat a introduir seria aquesta: oferir l'oportunitat als usuaris de poder donar la seva honesta opinió respecte a l'aplicació, a fi de corregir bugs i oferir a l'usuari el màxim confort possible.

En segon lloc és important deixar als usuaris reportar anuncis mal categoritzats, que mostrin animals en condicions pobres. A més, aquesta funcionalitat és essencial per evitar el tràfic d'animals a través de l'aplicació.

En tercer lloc, el xat en temps real de l'aplicació és molt bàsic, tot i que és suficient. Per aquesta raó es creu que es podrien afegir noves funcionalitats per permetre eliminar, editar i contestar explícitament missatges.

Per acabar, el mapa dinàmic que mostra la localització de les protectores més properes podria aportar més informació si aquest sigues més navegacional.

11 Glossari

En aquest apartat es mostren preguntes i respostes que aclareixen conceptes o termes els quals són necessaris conèixer per entendre a la perfecció el desenvolupament d'aquest projecte.

11.1 Què és una aplicació o *software*?

Encara que una aplicació es defineix com un conjunt d'ordres que es dona a un ordinador per realitzar una tasca determinada, en aquest projecte s'associarà el terme aplicació a aplicació mòbil.

11.2 Què és front-end i back-end?

És important aclarir aquests dos conceptes, ja que es repeteixen sovint en la documentació d'aquest projecte.

Front-end és la part de l'aplicació o software més propera a l'usuari – en aquest cas l'aplicació mòbil-, mentre que el terme back-end es refereix a la part més llunyana – en aquest cas el servidor-.

11.3 Què és una API?

Les sigles API venen de "*Application Program Interface*"; bàsicament és un conjunt de rutines, protocols, i eines per construir aplicacions.

En el marc d'aquest treball, es desenvoluparà una API, que més endavant es farà servir per desenvolupar el front-end de l'aplicació.

11.4 Què és UI i UX? [36]

Aquests dos conceptes juguen un paper important en el desenvolupament del front-end de l'aplicació:

Primerament, UI són les sigles per a User Interface, el que es tradueix com a interfície d'usuari. L'objectiu de la UI és guiar a l'usuari per l'aplicació durant el temps que aquest la fa servir.

D'altra banda, UX són les sigles per a User Experience, que es tradueix com a experiència d'usuari. L'objectiu d'aquesta és minimitzar la dificultat d'interacció entre l'usuari i la UI.

11.5 Què és un Framework?

Un Framework és un entorn de treball. Per concretar, en el món informàtic es refereix a Framework com a un conjunt de pautes, models i/o valors que ofereixen una estructura o bones pràctiques en el procés de desenvolupament d'un sistema, alhora que el facilita.

11.6 Què és Flutter?

Flutter [36] és un Framework desenvolupat per Google, que ajuda a crear aplicacions multiplataforma per a Android³¹, iOS³², i pròximament per a Web i escriptori. Aquest entorn presumeix d'oferir una gran expressivitat i flexibilitat de la UI, eficiència d'aplicació nativa, i a més un desenvolupament fluid i ràpid.

11.7 Què és PetsWorld?

PetsWorld és un portal d'anuncis digital sobre tota mena d'articles i serveis relacionats amb mascotes, on a més a més es podrà adoptar-ne. Tot i que aquesta plataforma serà desplegada més endavant també per a web, en aquest projecte només s'ha desenvolupat l'aplicació mòbil, per a Android i iOS, i el back-end corresponent per gestionar la base de dades.

12 Referències

- [1] "UPC - UPC Universitat Politècnica de Catalunya." <https://www.upc.edu/ca> (accessed Feb. 19, 2020).
- [2] "Resultados del Estudio del abandono en España 2018 | Fundación Affinity." <https://www.fundacion-affinity.org/blog/resultados-del-estudio-del-abandono-en-espana-2018> (accessed Feb. 22, 2020).
- [3] "Casi la mitad de animales abandonados en España fueron adoptados en 2018." <https://www.20minutos.es/noticia/3707437/0/total-138000-perros-gatos-abandonados-esapna-2018-asociacion-espanola-veterinarios/> (accessed Feb. 22, 2020).
- [4] "La venta ilegal de animales por internet: un mercado sin control que mueve más de mil millones de euros en Europa." https://www.eldiario.es/catalunya/Venta-animales-internet_0_936506491.html (accessed Feb. 22, 2020).
- [5] "Acoge Un Perro - Aplicaciones en Google Play." <https://play.google.com/store/apps/details?id=com.adopt.adoptaapi> (accessed Feb. 23, 2020).
- [6] "Miwuki Pet Shelter - Adopción de perros y gatos - Aplicaciones en Google Play." <https://play.google.com/store/apps/details?id=com.miwuki.petshelter> (accessed Feb. 23, 2020).

³¹ Sistema operatiu mòbil basat en el nucli Linux, desenvolupat per Google

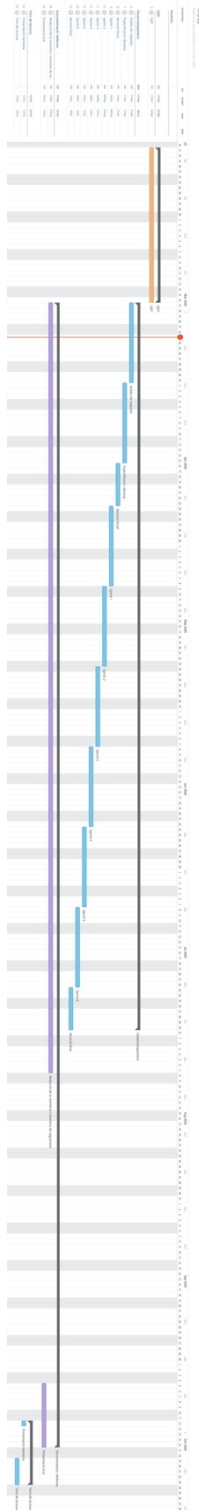
³² Sistema operatiu mòbil desenvolupat per Apple Inc.

- [7] "Chuby - adopta un perro, gato y más mascotas! - Aplicaciones en Google Play." <https://play.google.com/store/apps/details?id=rocks.chuby> (accessed Feb. 23, 2020).
- [8] "Appets, Adopta una mascota - Aplicaciones en Google Play." <https://play.google.com/store/apps/details?id=co.applist.appets> (accessed Feb. 23, 2020).
- [9] "Design - Material Design." <https://material.io/design/> (accessed Feb. 23, 2020).
- [10] "Express - Infraestructura de aplicaciones web Node.js." <https://expressjs.com/es/> (accessed Feb. 23, 2020).
- [11] "Node.js." <https://nodejs.org/es/> (accessed Feb. 23, 2020).
- [12] "La base de datos líder del mercado para aplicaciones modernas | MongoDB." <https://www.mongodb.com/es> (accessed Feb. 23, 2020).
- [13] "GraphQL | A query language for your API." <https://graphql.org/> (accessed Feb. 23, 2020).
- [14] "Por que Agile tiene cuatro veces mas éxito que Waterfall." <https://jeronimopalacios.com/agile/agile-tiene-cuatro-veces-mas-posibilidades-de-exito-que-waterfall/> (accessed Mar. 21, 2020).
- [15] "Manifiesto ágil - Wikipedia, la enciclopedia libre." https://es.wikipedia.org/wiki/Manifiesto_%C3%A1gil (accessed Mar. 21, 2020).
- [16] "What Is Kanban? An Overview Of The Kanban Method." <https://www.digite.com/kanban/what-is-kanban/> (accessed Feb. 23, 2020).
- [17] "Scrum (desarrollo de software) - Wikipedia, la enciclopedia libre." [https://es.wikipedia.org/wiki/Scrum_\(desarrollo_de_software\)](https://es.wikipedia.org/wiki/Scrum_(desarrollo_de_software)) (accessed Mar. 16, 2020).
- [18] "What Is Kanban? An Overview Of The Kanban Method." <https://www.digite.com/kanban/what-is-kanban/> (accessed Mar. 16, 2020).
- [19] "Git." <https://git-scm.com/> (accessed Feb. 23, 2020).
- [20] "Free Git GUI Client - Windows, Mac & Linux | GitKraken." <https://www.gitkraken.com/> (accessed Feb. 25, 2020).
- [21] "GitHub." <https://github.com/> (accessed Feb. 25, 2020).
- [22] "Issue Board - Task Tracking | GitKraken Glo Boards." <https://www.gitkraken.com/glo> (accessed Feb. 25, 2020).

- [23] “Ofertas de trabajo, bolsa de trabajo | Buscar empleo en Indeed España.” <https://www.indeed.es/?from=gnav-employer--tophat--jobseeker> (accessed Mar. 04, 2020).
- [24] “(No Title).” <https://www.fdi.ucm.es/profesor/gmendez/docs/is0809/ieee830.pdf> (accessed Sep. 30, 2020).
- [25] “Bloc.” <https://bloclibrary.dev/#/> (accessed Sep. 27, 2020).
- [26] “ArticleS.UncleBob.PrinciplesOfOod.” <http://butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod> (accessed Apr. 23, 2020).
- [27] “Mongoose ODM v5.10.8.” <https://mongoosejs.com/> (accessed Oct. 06, 2020).
- [28] “The Apollo platform - Apollo Basics - Apollo GraphQL Docs.” <https://www.apollographql.com/docs/intro/platform/> (accessed Oct. 08, 2020).
- [29] “Meet Google Play’s target API level requirement.” <https://developer.android.com/distribute/best-practices/develop/target-sdk> (accessed Oct. 01, 2020).
- [30] “Announcing Flutter 1.20: Faster, More Widgets, Improved Tooling | Flutter.” <https://medium.com/flutter/announcing-flutter-1-20-2aaf68c89c75> (accessed Oct. 01, 2020).
- [31] “built_value | Dart Package.” https://pub.dev/packages/built_value (accessed Oct. 01, 2020).
- [32] “google_maps_flutter | Flutter Package.” https://pub.dev/packages/google_maps_flutter (accessed Oct. 01, 2020).
- [33] “geocoder | Flutter Package.” <https://pub.dev/packages/geocoder> (accessed Oct. 01, 2020).
- [34] “Paginación: Qué es y cómo implementarlo correctamente utilizando 4 métodos.” <https://sitechecker.pro/es/pagination/> (accessed Oct. 01, 2020).
- [35] “graphql/graphql-playground: 🎮 GraphQL IDE for better development workflows (GraphQL Subscriptions, interactive docs & collaboration).” <https://github.com/graphql/graphql-playground> (accessed Oct. 05, 2020).
- [36] “Flutter - Beautiful native apps in record time.” <https://flutter.dev/> (accessed Feb. 22, 2020).
- [37] “GraphQL Cursor Connections Specification.” [https://relay.dev/graphql/connections.htm#ApplyCursorsToEdges\(\)](https://relay.dev/graphql/connections.htm#ApplyCursorsToEdges()) (accessed Oct. 06, 2020).

- [38] “Relay · The production-ready GraphQL client for React.” <https://relay.dev/> (accessed Oct. 06, 2020).
- [39] “DigitalOcean – The developer cloud.” <https://www.digitalocean.com/> (accessed Oct. 06, 2020).
- [40] “The color system - Material Design.” <https://material.io/design/color/the-color-system.html> (accessed Oct. 09, 2020).

Annex 1. Diagrama de Gantt



Annex 2. Índex de taules

Taula 1: Comparativa entre aplicacions existents i PetsWorld	12
Taula 2: Etapes del desenvolupament	23

Taula 3: Gestió dels riscos	24
Taula 4: Costos directes per activitat	24
Taula 5: Contingències	25
Taula 6: Pressupost final	25
Taula 7: Descripció del cas d'ús "Registrar-se"	28
Taula 8: Descripció del cas d'ús "Iniciar sessió"	28
Taula 9: Descripció del cas d'ús "Modificar perfil"	29
Taula 10: Descripció del cas d'ús "Consultar perfil propi o extern"	29
Taula 11: Descripció del cas d'ús "Enviar un missatge a un altre usuari"	30
Taula 12: Descripció del cas d'ús "Consultar anuncis"	30
Taula 13: Descripció del cas d'ús "Cerca anuncis per filtres"	31
Taula 14: Descripció del cas d'ús "Cerca protectores per proximitat "	31
Taula 15: Descripció del cas d'ús "Valorar perfil extern "	32
Taula 16: Descripció del cas d'ús "Crear o eliminar anuncis de mascotes"	33
Taula 17: Descripció del cas d'ús "Crear o eliminar anuncis de serveis"	33
Taula 18: Descripció del cas d'ús "Crear o eliminar anuncis d'articles"	34
Taula 19: Descripció del cas d'ús "Guardar anuncis a preferits"	34
Taula 20: Descripció del cas d'ús "Eliminar anuncis de preferits"	35
Taula 21: Taula de Volere del requisit r015	35
Taula 22: Taula de Volere del requisit R016	36
Taula 23: Taula de Volere del requisit R017	36
Taula 24: Taula de Volere del requisit R018	36
Taula 25: Taula de Volere del requisit R019	37
Taula 26: Taula de Volere del requisit R020	37
Taula 27: Taula de Volere del requisit R021	37
Taula 28: Taula de Volere del requisit R022	37
Taula 29: Taula de Volere del requisit R023	38
Taula 30: Taula de Volere del requisit R024	38
Taula 31: Distribució de la versió d'android en dispositius, a data de 10 d'abril de 2020	66
Taula 32: VALIDACIÓ DE LA FUNCIONALITAT REGISTRAR-SE	78
Taula 33: Validació de la funcionalitat inici de sessió	79
Taula 34: Validació de la funcionalitat modificar perfil.....	79
Taula 35: Validació de la funcionalitat consultar perfil propi o extern	79
Taula 36: Validació de la funcionalitat enviar un missatge a un altre usuari	80
Taula 37: Validació de la funcionalitat consultar anuncis	80
Taula 38: Validació de la funcionalitat cercar anuncis per filtres.....	80

Taula 39: Validació de la funcionalitat cercar protectores per proximitat	80
Taula 40: Validació de la funcionalitat crear, actualitzar o eliminar una valoració a un perfil extern	81
Taula 41: Validació de la funcionalitat crear o eliminar anuncis de mascotes.....	81
Taula 42: Validació de la funcionalitat crear o eliminar anuncis de serveis.....	81
Taula 43: Validació de la funcionalitat crear o eliminar anuncis d'articles	82
Taula 44: Validació de la funcionalitat guardar anuncis a preferits	82
Taula 45: Validació de la funcionalitat eliminar anuncis de preferits	82
Taula 46: Etapes del desenvolupament	88
Taula 47: Costos directes finals per rols.....	89
Taula 48: Matriu de sostenibilitat del TFG	92

Annex 3. Índex de il·lustracions

Il·lustració 1: GRÀFICA, EN CASTELLÀ, SOBRE EL DESTÍ DELS ANIMALS ADOPTATS EN EL 2017 [2] ..	10
Il·lustració 2: Model cascada - elaboració pròpia	17
Il·lustració 3: EXEMPLE D'UN TAULELL KANBAN [18].....	18
Il·lustració 4: Diagrama genèric de casos d'ús – elaboració pròpia	27
Il·lustració 5: Esquema conceptual de dades del sistema.....	39
Il·lustració 6: Abstracció de l'arquitectura del sistema	42
Il·lustració 7: Arquitectura lògica de tres capes.....	43
Il·lustració 8: Abstracció del patró BLoC, utilitzat en el front-end	44
Il·lustració 9: Diagrama del patró repositori	45
Il·lustració 10: Abstracció de l'arquitectura lògica del back-end, basada en capes	46
Il·lustració 11: Arquitectura lògica i física del sistema complet.....	47
Il·lustració 12: Pantalla principal	48
Il·lustració 13: Pantalla d'anunci	49
Il·lustració 14: Perfil d'usuari	50
Il·lustració 15: Inici de sessió i registre.....	51
Il·lustració 16: Pantalles de preferits, de xats, i d'un sol xat.....	52
Il·lustració 17: Pantalles de pujar anunci i d'editar perfil.....	53
Il·lustració 18: Pantalla d'opcions de l'aplicació i pantalla principal amb tema fosc	54
Il·lustració 19: Mapa navegacional de les diferents pantalles de l'aplicació.....	55
Il·lustració 20: Diagrama de classes del registre en el front end	56
Il·lustració 21: Diagrama de seqüència de la funcionalitat registrar-se en el frontend	57
Il·lustració 22: Diagrama de classes del back-end.....	61
Il·lustració 23: Diagrama de seqüència del back-end, relatiu a la funcionalitat de registrar-se	62

Il·lustració 24: Composició jeràrquica de widgets, font: flutter.dev	67
Il·lustració 25: Demostració gràfica del mapa dinàmic de l'aplicació	68
Il·lustració 26: Exemple d'un índex de pàgines [34].....	69
Il·lustració 27: Demostració dels marcadors de contingut en l'aplicació.....	70
Il·lustració 28: Possibles aparences de l'aplicació	71
Il·lustració 29: GraphQL IDE Playground	72
Il·lustració 30: Gràfica comparativa entre la paginació basada en cursors i la basada en offset.....	74
Il·lustració 31: Validació del requisit r015.....	83
Il·lustració 32: Validació del requisit r016.....	83
Il·lustració 33: Validació del requisit r017	83
Il·lustració 34: Validació del requisit r018.....	84
Il·lustració 35: Validació del requisit r019.....	84
Il·lustració 36: Validació del requisit r020.....	84
Il·lustració 37: Validació del requisit r021.....	84
Il·lustració 38: Validació del requisit r022.....	85
Il·lustració 39: Validació del requisit r023.....	85
Il·lustració 40: Validació del requisit r024.....	85